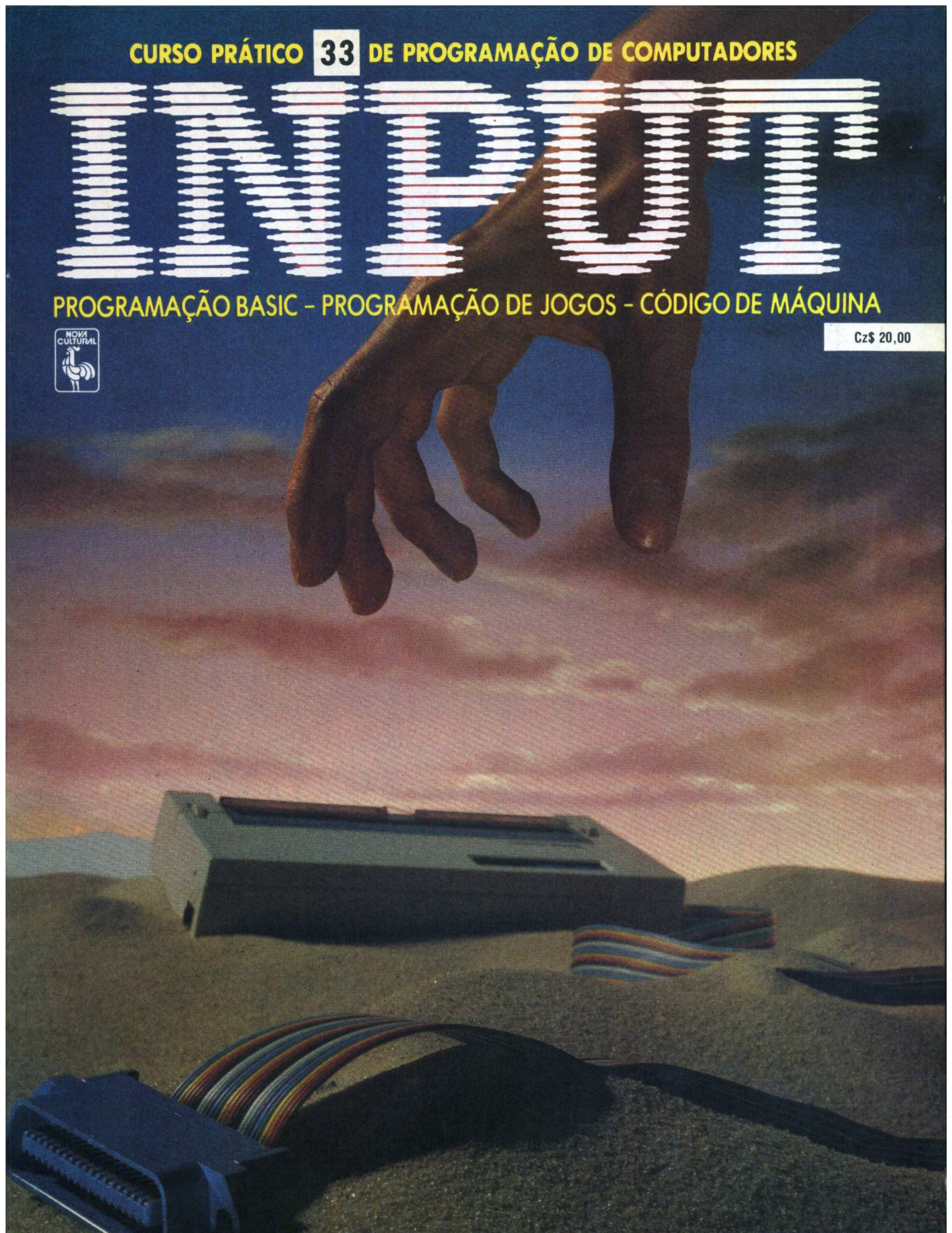
CURSO PRÁTICO 33 DE PROGRAMAÇÃO DE COMPUTADORES

INPUT

PROGRAMAÇÃO BASIC - PROGRAMAÇÃO DE JOGOS - CÓDIGO DE MÁQUINA

NOVA CULTURAL

Cz$ 20,00



INPUT

Vol. 3 Nº 33

NESTE NÚMERO

PROGRAMAÇÃO BASIC

PROGRAMAÇÃO DE GRÁFICOS EM 3-D (3)

Os três pontos de fuga. Observe um cubo a partir de vários ângulos 641

PERIFÉRICOS

CONECTE UMA IMPRESSORA

Conheça as condições para que uma impressora seja conectada ao computador 648

PROGRAMAÇÃO DE JOGOS

FELIZ ATERRISSAGEM!

Desligue o piloto automático e faça o avião aterrissar. Mas cuidado com os urubus 653

CÓDIGO DE MÁQUINA

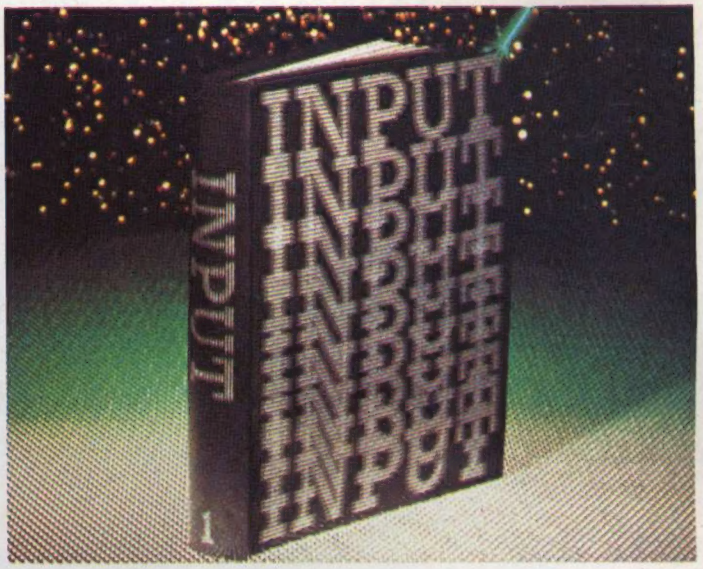
UM RELÓGIO NA TELA

Crie um relógio digital na tela e rompa o tédio de uma tarde de chuva 658

PROGRAMAÇÃO BASIC

SPRITES PARA O TRS-80 (2)

Aprenda a utilizar os caracteres gráficos pré-programados do TRS-80 660



PLANO DA OBRA

"INPUT" é uma obra editada em fascículos semanais, e cada conjunto de 15 fascículos compõe um volume. A capa para encadernação de cada volume estará à venda oportunamente.

COMPLETE SUA COLEÇÃO

Exemplares atrasados, até seis meses após o encerramento da coleção, poderão ser comprados, a preços atualizados, da seguinte forma: **1. Pessoalmente** — por meio de seu jornaleiro ou dirigindo-se ao distribuidor local, cujo endereço poderá ser facilmente conseguido junto a qualquer jornaleiro de sua cidade. Em São Paulo os endereços são: Rua Brigadeiro Tobias, 773 (Centro); Av. Industrial, 117 (Santo André); e, no Rio de Janeiro: Rua da Passagem, 93 (Botafogo). **2. Por carta** — Poderão ser solicitados exemplares atrasados também por carta, que deve ser enviada para DINAP — Distribuidor Nacional de Publicações — Números Atrasados — Estrada Velha de Osasco, 132 (Jardim Tereza) — CEP 06000 — Osasco — São Paulo. **3. Por telex** — Utilize o nº (011) 33670 ABSA. Em **Portugal**, os pedidos devem ser feitos à Distribuidora Jardim de Publicações Ltd. — Qta. Pau Varais, Azinhaga de Fetais — 2685, Camarate — Lisboa; Tel. 257-2542 — Apartado 57 — Telex 43 069 JARLIS P.
Não envie pagamento antecipado. O atendimento será feito pelo reembolso postal e o pagamento, incluindo as despesas postais, deverá ser efetuado ao se retirar a encomenda na Agência do Correio. **Atenção**: Após seis meses do encerramento da coleção, os pedidos serão atendidos, dependendo da disponibilidade de estoque. **Obs.:** Quando pedir livros, mencione sempre o título e/ou o autor da obra, além do número da edição.

COLABORE CONOSCO

Encaminhe seus comentários, críticas, sugestões ou reclamações ao Serviço de Atendimento ao Leitor — Caixa Postal 9442, São Paulo — SP.

Editor
VICTOR CIVITA

REDAÇÃO
Diretora Editorial: Iara Rodrigues

Editor Executivo: Antonio José Filho

Editor Chefe: Paulo de Almeida
Editor de Texto: Cláudio A. V. Cavalcanti
Chefe de Arte: Carlos Luiz Batista
Assistentes de Arte: Ailton Oliveira Lopes, Dilvacy M. Santos, Grace Alonso Arruda, José Maria de Oliveira, Monica Lenardon Corradi
Secretária de Redação/Coordenadora: Stefania Crema
Secretários de Redação: Beatriz Hagström, José Benedito de Oliveira Damião, Maria de Lourdes Carvalho, Marisa Soares de Andrade, Mauro de Queiroz

COLABORADORES
Consultor Editorial Responsável: Dr. Renato M. E. Sabbatini (Diretor do Núcleo de Informática Biomédica da Universidade Estadual de Campinas)
Execução Editorial: DATAQUEST Assessoria em Informática Ltda., Campinas, SP
Tradução: Reinaldo Cúrcio
Tradução, adaptação, programação e redação: Abílio Pedro Neto, Aluísio J. Dornellas de Barros, Marcelo R. Pires Therezo, Raul Neder Porrelli
Coordenação Geral: Rejane Felizatti Sabbatini
Editora de Texto: Ana Lúcia B. de Lucena
Assistente de Arte: Dagmar Bastos Sampaio

COMERCIAL
Diretor Comercial: Roberto Martins Silveira
Gerente Comercial: Flávio Maculan
Gerente de Circulação: Denise Maria Mozol

PRODUÇÃO
Gerente de Produção: João Stungis
Coordenador de Impressão: Atílio Roberto Bonon

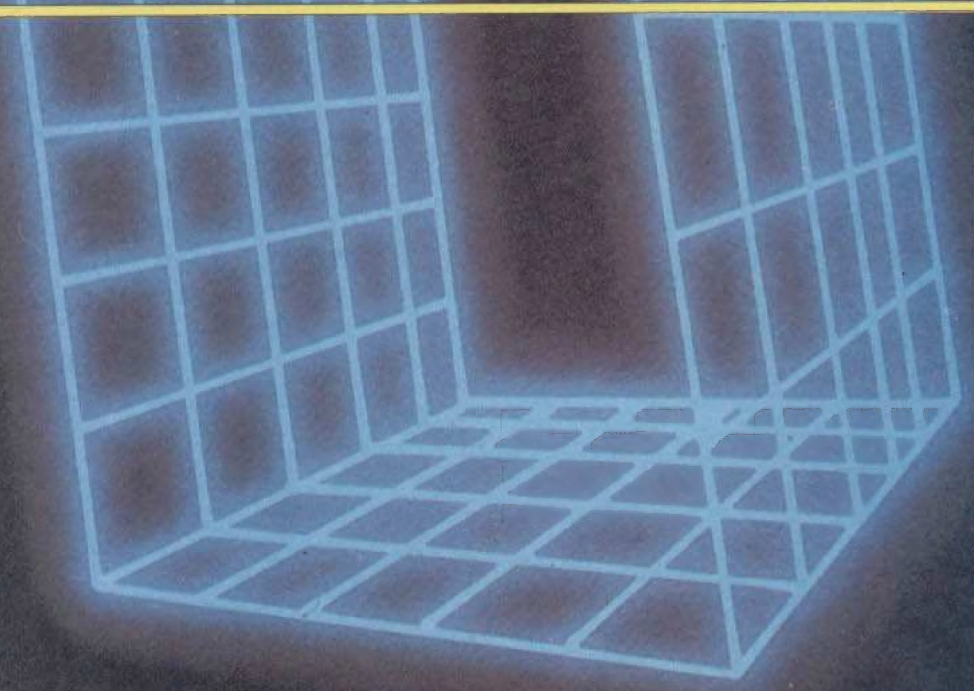
Preparador de Texto/Coordenador: Eliel Silveira Cunha
Preparadores de Texto: Alzira Moreira Braz, Ana Maria Dilguerian, Karina Ap. V. Grechi, Levon Yacubian, Luciano Tasca, Maria Teresa Galluzzi, Maria Teresa Martins Lopes, Paulo Felipe Mendrone
Revisor/Coordenador: José Maria de Assis
Revisoras: Conceição Aparecida Gabriel, Isabel Leite de Camargo, Ligia Aparecida Ricetto, Maria de Fátima Cardoso, Nair Lúcia de Brito
Paste-up: Anastase Potaris, Balduino F. Leite, Edson Donato

© Marshall Cavendish Limited, 1984/85.
© Editora Nova Cultural Ltda., São Paulo, Brasil, 1986.
Edição organizada pela Editora Nova Cultural Ltda.
Av. Brigadeiro Faria Lima, nº 2000 - 3º andar
CEP 01452 - São Paulo - SP - Brasil
(Artigo 15 da Lei 5 988, de 14/12/1973).
Esta obra foi composta na AM Produções Gráficas Ltda.
e impressa na Divisão Gráfica da Editora Abril S.A.

PROGRAMAÇÃO DE GRÁFICOS EM 3-D (3)

- O PONTO DE OBSERVAÇÃO
- ANTES DE COMEÇAR
- AS TRANSFORMAÇÕES
- INICIALIZE AS VARIÁVEIS
- COMO CHAMAR A ROTINA

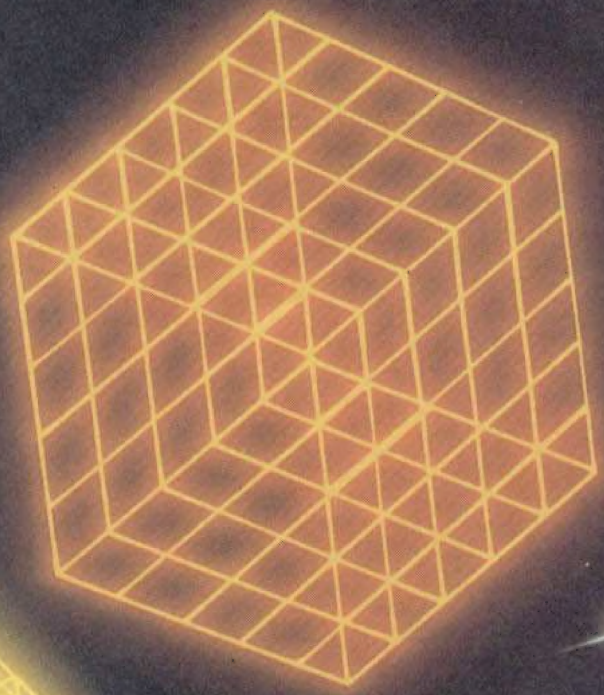
As leis da perspectiva foram formuladas no decorrer do Renascimento por artistas como Alberti e Leonardo. Aqui, você poderá aplicá-las, acionando apenas algumas teclas.

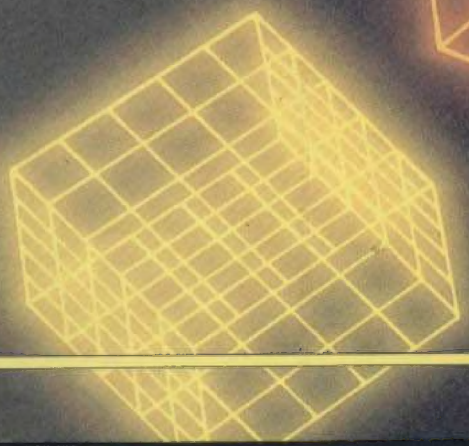


Embora nos permitisse desenhar um cubo em três dimensões e variar o seu tamanho, o programa apresentado nos dois últimos artigos desta série tinha uma limitação: com ele era possível visualizar o desenho apenas de frente. Assim, era impossível observá-lo de cima, de lado ou de qualquer outro ângulo que desejássemos. Este artigo oferece rotinas extras que nos permitirão especificar a posição dos nossos olhos, de modo a poder observar o cubo de qualquer posição.

Essa flexibilidade é muito útil, sobretudo para desenhos de objetos mais complicados, pois a visão que obtemos nas diferentes posições pode nos revelar traços escondidos ou obscuros. Ao especificarmos uma sucessão de coordenadas diferentes para a posição dos nossos olhos, teremos a impressão de estar "sobrevoando" o objeto.

No entanto, embora usemos os mesmos princípios, a seqüência de desenhos que pode ser gerada em um computador pessoal é muito lenta e o efeito obtido, inferior ao dos desenhos tipo *wireframe* dos comerciais de televisão. Nestes últimos, temos a impressão de que o observador aproxima-se de um objeto (carro, nave espacial etc.) e o contorna a uma alta velocidade.



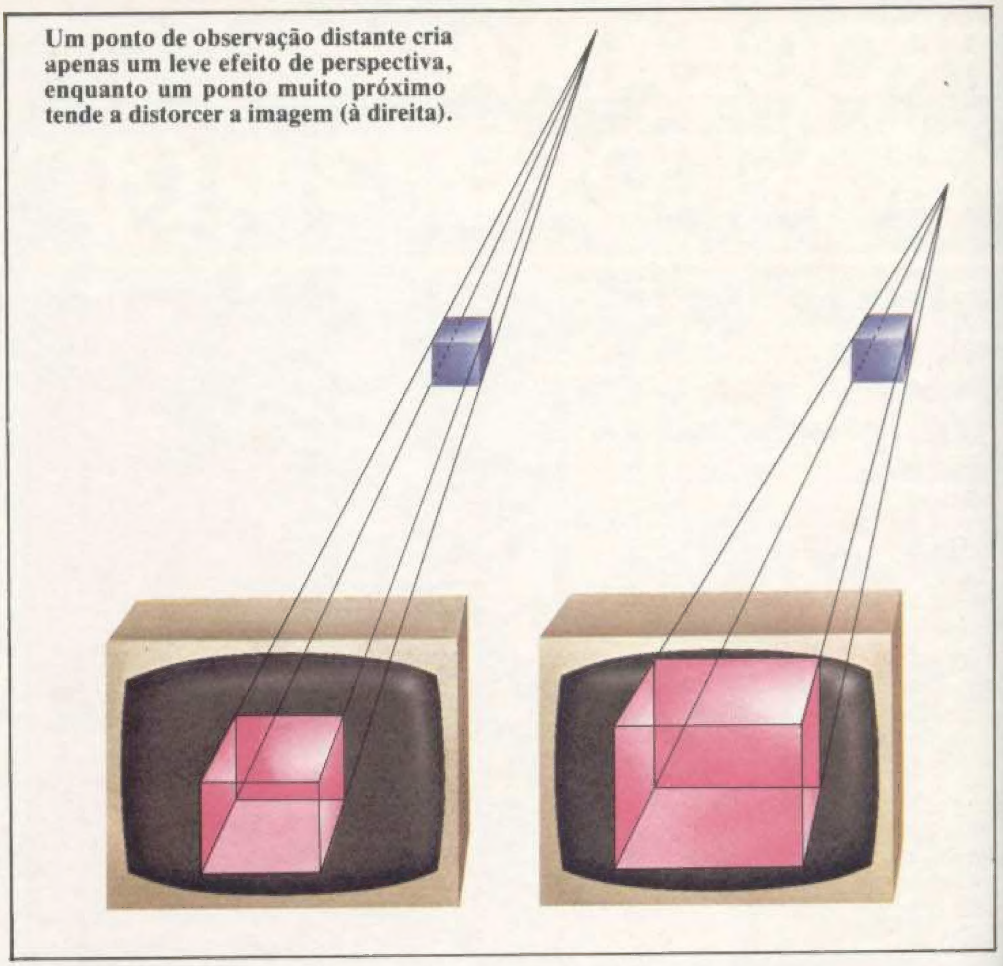


O EFEITO DO PONTO DE OBSERVAÇÃO

Já mostramos como é possível representar um objeto tridimensional em uma tela de duas dimensões, usando convenções visuais que serão interpretadas pelos olhos e pelo cérebro do observador. Os programas anteriores fizeram isso por intermédio de uma projeção isométrica, na qual linhas oblíquas são vistas no contexto do desenho entrando ou saindo da tela.

A forma mais comum de convenção visual é o desenho em perspectiva, em que as linhas que se afastam do observador (ou seja, "entram" na tela) convergem para um ponto distante, conhecido como Ponto de Fuga, e têm diminuídas as suas dimensões, à medida que se afastam do primeiro plano. Na realidade há três pontos de fuga na perspectiva em 3-D: um para cada linha ortogonal desenhada ao longo dos eixos. A posição dos três pontos é fixada com base na relação entre as posições do observador e as do objeto a ser representado (na convenção visual, esses pontos são imaginários).

Assim, a primeira coisa que precisaremos saber é a relação entre as posições do observador e as do objeto. Para determinarmos essa relação, devemos indicar o ponto de vista para os eixos X, Y e Z que foram especificados na tela. Os programas a seguir fazem exatamente isto, executando depois as transformações necessárias.

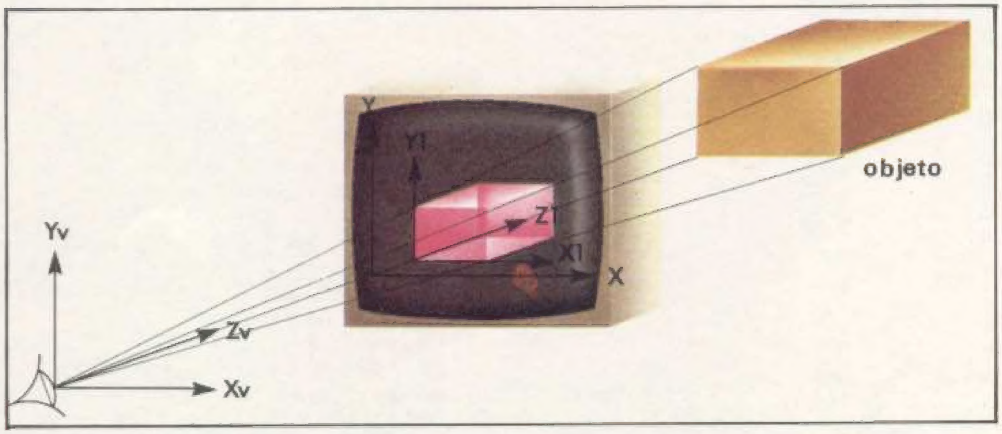


Um ponto de observação distante cria apenas um leve efeito de perspectiva, enquanto um ponto muito próximo tende a distorcer a imagem (à direita).

ANTES DE COMEÇARMOS

Para rodar esses novos segmentos de programa listados abaixo, recorremos à Rotina da Grade apresentada no artigo *Programação de Gráficos em 3-D (1)* à página 581. Essa rotina deve ser carregada no computador. Uma boa idéia é carregar também a Rotina do Desenho do Círculo que se encontra no mesmo artigo. Embora esta última não seja necessária por enquanto, convém termos todas as rotinas juntas, prontas para o programa global de desenhos que sairá no próximo artigo. Desse modo, as linhas que precisaremos agora são as que vão de 5000 a 6160; todas as outras podem ser apagadas.

Como anteriormente, teremos que entrar várias porções de programa antes de podermos rodar qualquer coisa, pois a posição dos olhos, da mesma maneira que as transformações para rotação e perspectiva, devem ser implementadas em primeiro lugar.



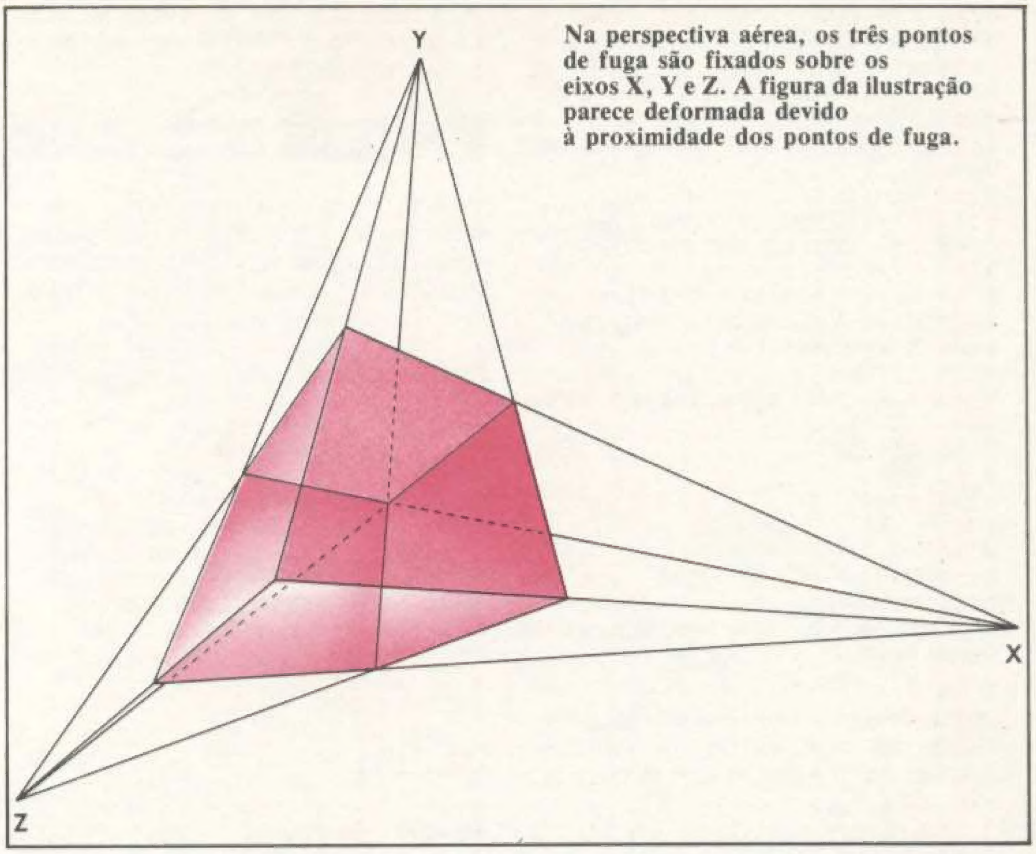
DEFINIÇÃO DO PONTO DE OBSERVAÇÃO

A primeira parte do programa calcula as variáveis necessárias para se determinar a posição do olho do observador no espaço tridimensional:

```
8000 XV=X:YV=Y:ZV=Z
8010 WV=YV*YV+ZV*ZV
8020 PV=SQR(XV*XV+WV)
8030 IF PV=0 THEN RETURN
8035 WV=SQR(WV)
8040 XU=XV/PV
8050 YU=YV/PV
8060 ZU=ZV/PV
8070 WU=WV/PV
8080 REM ORIENTAÇÃO DO OLHO
8090 A=XV*YV:B=ZV:GOSUB 8450:G=
H
8100 A=YV:B=XV:GOSUB 8450:G=G+H
8110 SG=SIN(G)
8120 CG=COS(G)
8130 REM MATRIZ DE ROTAÇÃO
8140 R1=WU*CG
8150 R2=-WU*SG
8160 R3=-XU
```

Para representar uma figura em perspectiva, é preciso transformar as coordenadas dos eixos tridimensionais (X1, Y1, Z1) em pontos referidos aos eixos da tela (X, Y).

```
8170 R4=-YU
8180 R5=-ZU
8190 R6=XV*XU+YV*YU+ZV*ZU
8200 IF WU=0 THEN 8340
8210 XT=XV*WU-(YV*YU+ZV*ZU)*XU/
WU
8220 YT=(YV*ZU-ZV*YU)/WU
8230 R7=(ZU*SG-XU*YU*CG)/WU
8240 R8=(-YU*SG-XU*ZU*CG)/WU
8250 R9=CG*XT+SG*YT
8260 S1=(ZU*CG+XU*YU*SG)/WU
8270 S2=(-YU*CG+XU*ZU*SG)/WU
8280 S3=-SG*XT+CG*YT
8330 RETURN
8340 REM CASO ESPECIAL NO EIXO
X
8350 R7=-1
8360 R8=0
8370 R9=0
8380 S1=0
8390 S2=1
8400 S3=0
8410 RETURN
8450 IF B<>0 THEN H=ATN(A/B):RE
TURN
8460 H=PI/2:RETURN
```



Na perspectiva aérea, os três pontos de fuga são fixados sobre os eixos X, Y e Z. A figura da ilustração parece deformada devido à proximidade dos pontos de fuga.

```
8000 XV = X:YV = Y:ZV = Z
8010 WV = YV * YV + ZV * ZV
8020 PV =  SQR (XV * XV + WV)
8030  IF PV = 0 THEN  RETURN
8035 WV =  SQR (WV)
8040 XU = XV / PV
8050 YU = YV / PV
8060 ZU = ZV / PV
8070 WU = WV / PV
8080  REM  ORIENTACAO DO OLHO
8090 A = XV * YV:B = ZV: GOSUB
8450:G = H
8100 A = YV:B = XV: GOSUB 8450:
G = G + H
8110 SG =  SIN (G)
8120 CG =  COS (G)
8130  REM  MATRIZ ROTACAO
8140 R1 = WU * CG
8150 R2 =  - WU * SG
8160 R3 =  - XU
8170 R4 =  - YU
8180 R5 =  - ZU
8190 R6 = XV * XU + YV * YU + Z
V * ZU
8200  IF WU = 0 THEN 8340
8210 XT = XV * WU - (YV * YU +
ZV * ZU) * XU / WU
8220 YT = (YV * ZU - ZV * YU) /
 WU
8230 R7 = (ZU * SG - XU * YU *
CG) / WU
8240 R8 = ( - YU * SG - XU * ZU
 * CG) / WU
8250 R9 = CG * XT + SG * YT
8260 S1 = (ZU * CG + XU * YU *
SG) / WU
8270 S2 = ( - YU * CG + XU * ZU
 * SG) / WU
8280 S3 =  - SG * XT + CG * YT
8330  RETURN
8340  REM  CASO ESPECIAL NO EIX
O-X
8350 R7 =  - 1
8360 R8 = 0
8370 R9 = 0
8380 S1 = 0
8390 S2 = 1
8400 S3 = 0
8410  RETURN
8450  IF B <  > 0 THEN H =  ATN
 (A / B): RETURN
8460 H = PI / 2: RETURN
```

```
8000 LET XV=X: LET YV=Y: LET ZV
=Z
8010 LET WV=YV*YV+ZV*ZV
8020 LET PV=SQR (XV*XV+WV)
8030 IF PV=0 THEN  RETURN
8035 LET WV=SQR WV
8040 LET XU=XV/PV
8050 LET YU=YV/PV
8060 LET ZU=ZV/PV
8070 LET WU=WV/PV
8080 REM ORIENTACAO
8090 LET A=XV*YV: LET B=ZV: GOS
UB 8450: LET G=H
8100 LET A=YV: LET B=XV: GOSUB
8450: LET G=G+H
8110 LET SG=SIN G
8120 LET CG=COS G
8140 LET R1=WU*CG
8150 LET R2=-WU*SG
8160 LET R3=-XU
8170 LET R4=-YU
8180 LET R5=-ZU
8190 LET R6=XV*XU+YV*YU+ZV*ZU
8200 IF WU=0 THEN  GOTO 8340
8210 LET XT=XV*WU-(YV*YU+ZV*ZU)
*XU/WU
8220 LET YT=(YV*ZU-ZV*YU)/WU
8230 LET R7=(ZU*SG-XU*YU*CG)/WU
8240 LET R8=(-YU*SG-XU*ZU*CG)/W
U
8250 LET R9=CG*XT+SG*YT
8260 LET S1=(ZU*CG+XU*YU*SG)/WU
8270 LET S2=(-YU*CG+XU*ZU*SG)/W
U
8280 LET S3=-SG*XT+CG*YT
8330 RETURN
8350 LET R7=-1
8360 LET R8=0
8370 LET R9=0
8380 LET S1=0
8390 LET S2=1
8400 LET S3=0
8410 RETURN
8450 IF B<>0 THEN  LET H=ATN (A
/B): RETURN
8460 LET H=PI/2: RETURN
```

```
8000 XV=X:YV=Y:ZV=Z
8010 WV=YV*YV+ZV*ZV
8020 PV=SQR(XV*XV+WV)
8030 IF PV=0 THEN RETURN
8035 WV=SQR(WV)
8040 XU=XV/PV
8050 YU=YV/PV
8060 ZU=ZV/PV
8070 WU=WV/PV
8080 REM ORIENTACAO
8090 A=XV*YV:B=ZV:GOSUB 8450:G=
H
8100 A=YV:B=XV:GOSUB 8450:G=G+H
8110 SG=SIN(G)
8120 CG=COS(G)
8130 REM ROTACAO DA MATRIZ
8140 R1=WU*CG
8150 R2=-WU*SG
```

```
8160 R3=-XU
8170 R4=-YU
8180 R5=-ZU
8190 R6=XV*XU+YV*YU+ZV*ZU
8200 IF WU=0 THEN 8340
8210 XT=XV*WU-(YV*YU+ZV*ZU)*XU/
WU
8220 YT=(YV*ZU-ZV*YU)/WU
8230 R7=(ZU*SG-XU*YU*CG)/WU
8240 R8=(-YU*SG-XU*ZU*CG)/WU
8250 R9=CG*XT+SG*YT
8260 S1=(ZU*CG+XU*YU*SG)/WU
8270 S2=(-YU*CG+XU*ZU*SG)/WU
8280 S3=-SG*XT+CG*YT
8330 RETURN
8340 REM CASO ESPECIAL NO EIXO
X
8350 R7=-1
8360 R8=0
8370 R9=0
8380 S1=0
8390 S2=1
8400 S3=0
8410 RETURN
8450 IF B<>0 THEN H=ATN(A/B) EL
SE H=PI/2
8460 RETURN
```

Para entendermos o que está acontecendo, devemos nos lembrar que o eixo Z é aquele que sai do centro da tela e vem na direção do observador; o eixo Y aponta para cima da tela e o eixo X, para os lados.

A posição dos olhos está em (**XV, YV, ZV**). A variável **WV** nos dá a distância nas direções Y e Z combinadas. Definimos a origem em (**0, 0, 0**) para a posição do objeto que observamos no espaço. Para simplificar, situamos essa posição no centro da tela. A linha 8030 interromperá a rotina, caso a posição dos olhos seja definida na origem. As variáveis **XU, YU, ZU** são as distâncias, na direção dos respectivos eixos (X, Y, X) de uma linha de comprimento unitário entre a origem e a posição dos olhos. **WU** é a distância nas direções Y e Z combinadas.

Se movermos o objeto da esquerda para a direita, nosso ângulo de visão sofrerá alterações. Esse ângulo é medido a partir do eixo X e definido nas linhas 8090 e 8100. Um teste é feito nas linhas 8450 e 8460 para evitar divisões por zero, que interromperiam o programa, gerando uma mensagem de erro.

Quando olhamos diretamente ao longo do eixo X, temos uma visão comum do objeto, mas, se rodarmos nosso ponto de observação, o objeto também rodará, proporcionando ângulos mais interessantes. As linhas 8140 a 8280 inicializam as variáveis que definem a orientação da posição do olho no espaço. Essa posição é tal que o seu eixo Z encontra-se ao longo da linha que vai do olho até a origem da tela.

O resto da rotina inicializa variáveis para casos especiais; por exemplo, quando a posição do olho estiver exatamente sobre o eixo X.

AS TRANSFORMAÇÕES

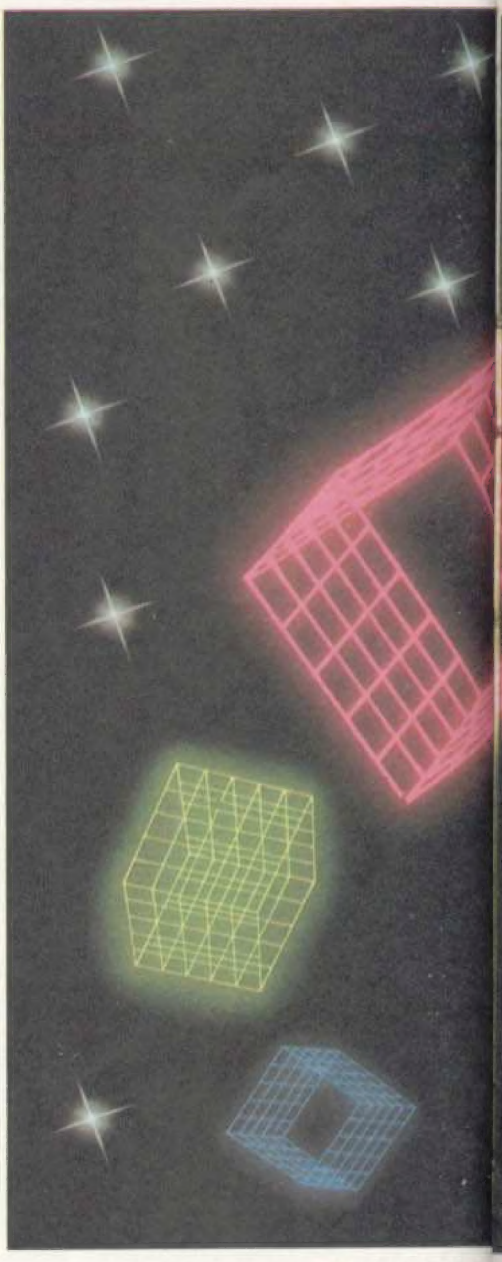
A próxima seção transforma as coordenadas X, Y e Z do cubo nas coordenadas finais da tela. Essa transformação leva em conta a posição do olho e o efeito da perspectiva.

```
8500 X1=T1*X+T4*Y+T7
8510 Y1=T2*X+T5*Y+T8
8520 Z1=T3*X+T6*Y+T9
8530 REM 2-D PARA 3-D
8540 X2=R1*X1+R7*Y1+R8*Z1+R9
8550 Y2=R2*X1+S1*Y1+S2*Z1+S3
8560 Z2=R3*X1+R4*Y1+R5*Z1+R6
8570 REM OBJETO PARA OLHO
8575 IF Z2<ZN THEN RETURN
8580 X3=D*X2/Z2
8590 Y3=D*Y2/Z2
8600 RETURN
```

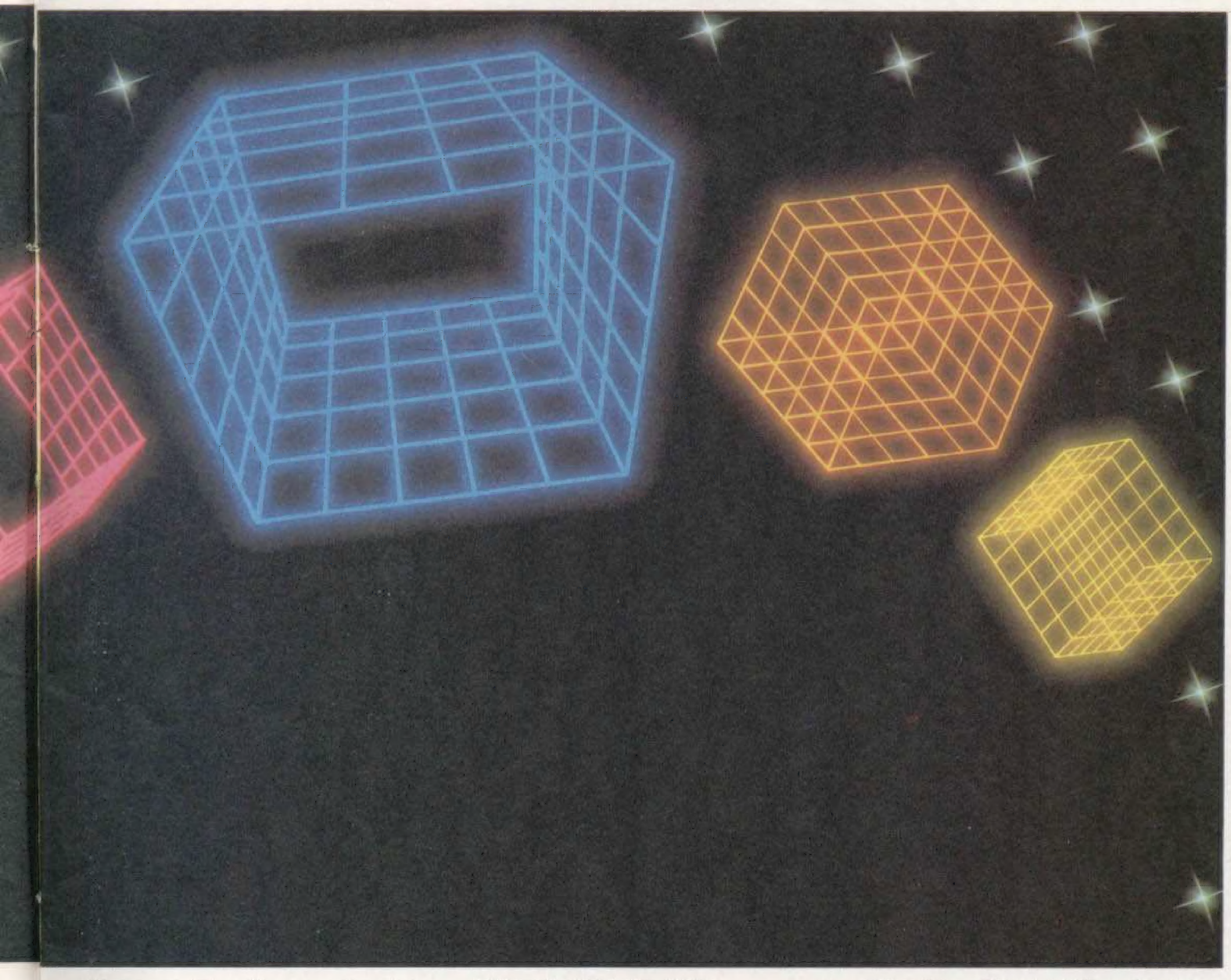
```
8500 X1 = T1 * X + T4 * Y + T7
8510 Y1 = T2 * X + T5 * Y + T8
8520 Z1 = T3 * X + T6 * Y + T9
8530  REM  2-D PARA 3-D
8540 X2 = R1 * X1 + R7 * Y1 + R
8 * Z1 + R9
8550 Y2 = R2 * X1 + S1 * Y1 + S
2 * Z1 + S3
8560 Z2 = R3 * X1 + R4 * Y1 + R
5 * Z1 + R6
8575  IF Z2 < ZN THEN  RETURN
8580 X3 = D * X2 / Z2
8590 Y3 =  - D * Y2 / Z2
8600  RETURN
```

```
8500 LET X1=T1*X+T4*Y+T7
8510 LET Y1=T2*X+T5*Y+T8
8520 LET Z1=T3*X+T6*Y+T9
8540 LET X2=R1*X1+R7*Y1+R8*Z1+R
9
8550 LET Y2=R2*X1+S1*Y1+S2*Z1+S
3
8560 LET Z2=R3*X1+R4*Y1+R5*Z1+R
6
8575 IF Z2<ZN THEN  RETURN
8580 LET X3=D*X2/Z2
8590 LET Y3=-D*Y2/Z2
8600 RETURN
```

Essa rotina usa uma matriz aritmética complicada para transformar as coordenadas (X, Y), que são basicamente os passos de transformação descritos no artigo anterior desta série — *Programação de Gráficos em 3-D (2)*. As linhas 8500 a 8520 transformam o plano cartesiano (X, Y) de duas dimensões (2-D) no espaço de três dimensões (3-D) (X1, Y1, Z1). As linhas 8540 a 8560 transformam as coodernadas espaciais (X1, Y1, Z1), de modo a posicioná-las de acordo com a localização do olho e a direção (X2, Y2, Z2).



A linha 8575 testa se a nova posição a ser encontrada está excessivamente próxima do olho (ou seja, se Z2 está entre 0 e ZN) ou se está situada atrás dos olhos. Em ambos os casos, a posição é ignorada, pois seria impossível visualizarmos o objeto. Se a posição estiver muito distante de ZN (a partir da posição em frente ao olho), então a localização das coodernadas da tela (X3, Y3) será calculada somente no fim da rotina. O parâmetro **D/Z2** nessas linhas acrescenta perspectiva à figura, projetando o objeto em uma tela plana a uma distância definida **D**. Se estabelecermos um valor pequeno para **D**, o efeito obtido será uma perspectiva mais nítida



(como se a tela estivesse mais próxima do usuário); se **D** for grande, a tela parecerá estar a uma grande distância do observador, de modo que o efeito de perspectiva será pequeno e a imagem aparecerá distorcida, mesmo quando vista obliquamente, de qualquer direção.

COMO INICIALIZAR AS VARIÁVEIS

O próximo passo será reescrever as Rotinas de Inicialização e Desenho para podermos usar as novas rotinas de transformação.

```
9000 CLS
9020 XM=256:YM=192
9030 XD=XM/2:YD=YM/2
9040 ZN=1
9042 CLS:INPUT"Distância do pla
no de projeção: ";D
9045 IF D<0 THEN D=1000*ZN
9050 T1=1:T2=0:T3=0
9060 T4=0:T5=1:T6=0
9070 T7=0:T8=0:T9=0
9085 CLS
9090 RETURN
9500 X=XS:Y=YS:GOSUB 8500:IF Z2
<ZN THEN 9520
9510 IX=128+X3:IY=97-Y3
9520 X=XE:Y=YE:GOSUB 8500:IF Z2
<ZN THEN 9550
9530 LINE(127+X3,95-Y3)-(IX,IY)
,15
9550 RETURN
```

```
9000  HOME
9020 XM = 280:YM = 192
9030 XD = XM / 2:YD = YM / 2
9040 ZN = 1
9042  INPUT "DISTANCIA AO PLANO
 DE PROJECAO ";D
9045  IF D < 0 THEN D = 1000 *
ZN
9050 T1 = 1:T2 = 0:T3 = 0
9060 T4 = 0:T5 = 1:T6 = 0
9070 T7 = 0:T8 = 0:T9 = 0
9090  HOME : RETURN
9500 X = XS:Y = YS: GOSUB 8500:
 IF Z2 < ZN THEN 9520
9505  IF X3 <  - 139 OR Y3 <  -
 95 OR X3 > 140 OR Y3 > 96 THEN
 9550
9510 IX = 139 + X3:IY = 95 + Y3

9520 X = XE:Y = YE: GOSUB 8500:
 IF Z2 < ZN THEN 9550
9525  IF X3 <  - 139 OR Y3 <  -
 95 OR X3 > 140 OR Y3 > 96 THEN
 9550
9530  HPLOT 139 + X3,95 + Y3 TO
```

```
 IX,IY
9550  RETURN
```

```
9000 CLS
9020 LET XM=256: LET YM=176
9030 LET XD=XM/2: LET YD=YM/2
9040 LET ZN=1
9042 INPUT "IMTRODUZA DISTANCIA
 AO PLANO DE PROJECAO",D
9045 IF D<=0 THEN  LET D=1000*Z
N
9050 LET T1=1: LET T2=0: LET T3
=0
9060 LET T4=0: LET T5=1: LET T6
=0
9070 LET T7=0: LET T8=0: LET T9
=0
9090 CLS : RETURN
9500 LET X=XS: LET Y=YS: GOSUB
8500: IF Z2<ZN THEN  GOTO 9520
9505 IF X3<-127 OR Y3<-87 OR X3
>128 OR Y3>88 THEN  GOTO 9550
9510 PLOT 127+X3,87+Y3
9520 LET X=XE: LET Y=YE: GOSUB
8500: IF Z2<ZN THEN  GOTO 9550
9525 IF X3<-127 OR Y3<-87 OR X3
>128 OR Y3>88 THEN  GOTO 9550
9530 DRAW 127+X3-PEEK 23677,87+
Y3-PEEK 23678
9550 RETURN
```

```
9000 PCLS
9020 XM=256:YM=192
9030 XD=XM/2:YD=YM/2
9040 ZN=1
9042 CLS:INPUT " INTRODUZA A DI
STANCIA AO PLANO  DE PROJECAO "
;D
9045 IF D<=0 THEN D=1000*ZN
9050 T1=1:T2=0:T3=0
9060 T4=0:T5=1:T6=0
9070 T7=0:T8=0:T9=0
9085 CLS
9090 RETURN
9500 X=XS:Y=YS:GOSUB 8500:IF Z2
<ZN THEN 9520
9505 IF X3<-127 OR Y3<-96 OR X3
>128 OR Y3>95 THEN 9550
9510 DRAW"BM"+STR$(INT(127+X3))
+","+STR$(INT(95-Y3))
9520 X=XE:Y=YE:GOSUB 8500:IF Z2
<ZN THEN  9550
9525 IF X3<-127 OR Y3<-96 OR X3
>128 OR Y3>95 THEN 9550
9530 LINE -(127+X3,95-Y3),PSET
9550 RETURN
```

A linha 9020 atribui às variáveis **XM** e **YM** as dimensões máximas da tela nas direções X e Y, respectivamente. A linha 9030 calcula o ponto médio; a 9040 calcula a posição mais próxima permitida nos pontos a serem desenhados, conforme a posição do olho.

A variável **D** nos dá a distância atual da posição do olho até o plano de projeção (a tela) e determina a perspectiva. Os valores de **D** são inseridos quando rodamos o programa, de modo que podemos variar o grau da perspectiva. Se nenhum valor for especificado e pressionarmos **ENTER** ou **RETURN** a linha 9045 adotará o valor 1000. As linhas 9050 a 9070 atribuem valores para as constantes de transformação, especificando o plano no qual será desenhada a imagem em perspectiva.

Esse trecho do programa continua com a Rotina de Desenho revisada. Na linha 9500, são inicializadas as variáveis para as constantes de transformação; nas linhas 9505 e 9525 é feito um teste para determinar se o novo ponto a ser desenhado (linhas 9510 e 9530) está dentro dos limites da tela. O MSX não precisa desse teste, pois ele ignora os pontos fora da tela. Nos outros micros, o teste serve para prevenir contra mensagens de erro.

CHAME A ROTINA

Precisaremos agora da rotina para desenhar a grade fornecida no primeiro artigo da série. Caso você não a tenha gravado, digite-a antes de inserir esta parte final do programa.

```
100 PI=4*ATN(1)
110 GOSUB 9000
120 L=20:N=5
125 GOSUB 505:GOTO 140
130 GOSUB 500
140 IF X=0 AND Y=0 AND Z=0 THEN
 170
150 GOSUB 1000
160 GOTO 130
170 CLS
180 END
500 IF INKEY$="" THEN 500
505 CLS
510 INPUT"Posição do observador
 (X,Y,Z): ";X,Y,Z
520 GOSUB 8000
530 CLS:SCREEN 2
540 RETURN
1000 P=L/2
1010 T1=1:T2=0:T3=0
1020 T4=0:T5=1:T6=0
1030 T7=-P:T8=-P:T9=-P
1040 GOSUB 1200 'BAIXO
1050 T7=-P:T8=-P:T9=P
1060 GOSUB 1200 'CIMA
1070 T4=0:T5=0:T6=-1
1080 GOSUB 1200 'ESQUERDA
1090 T7=-P:T8=P:T9=P
1100 GOSUB 1200 'DIREITA
1110 T1=0:T2=-1:T3=0
1120 GOSUB 1200 'TRAS
1130 T7=P:T8=P:T9=P
1140 GOSUB 1200 'FRENTE
1170 RETURN
1200 XA=0:YA=0:LW=L:LH=L:NX=N:N
Y=N
1210 GOSUB 5000
1220 RETURN
```

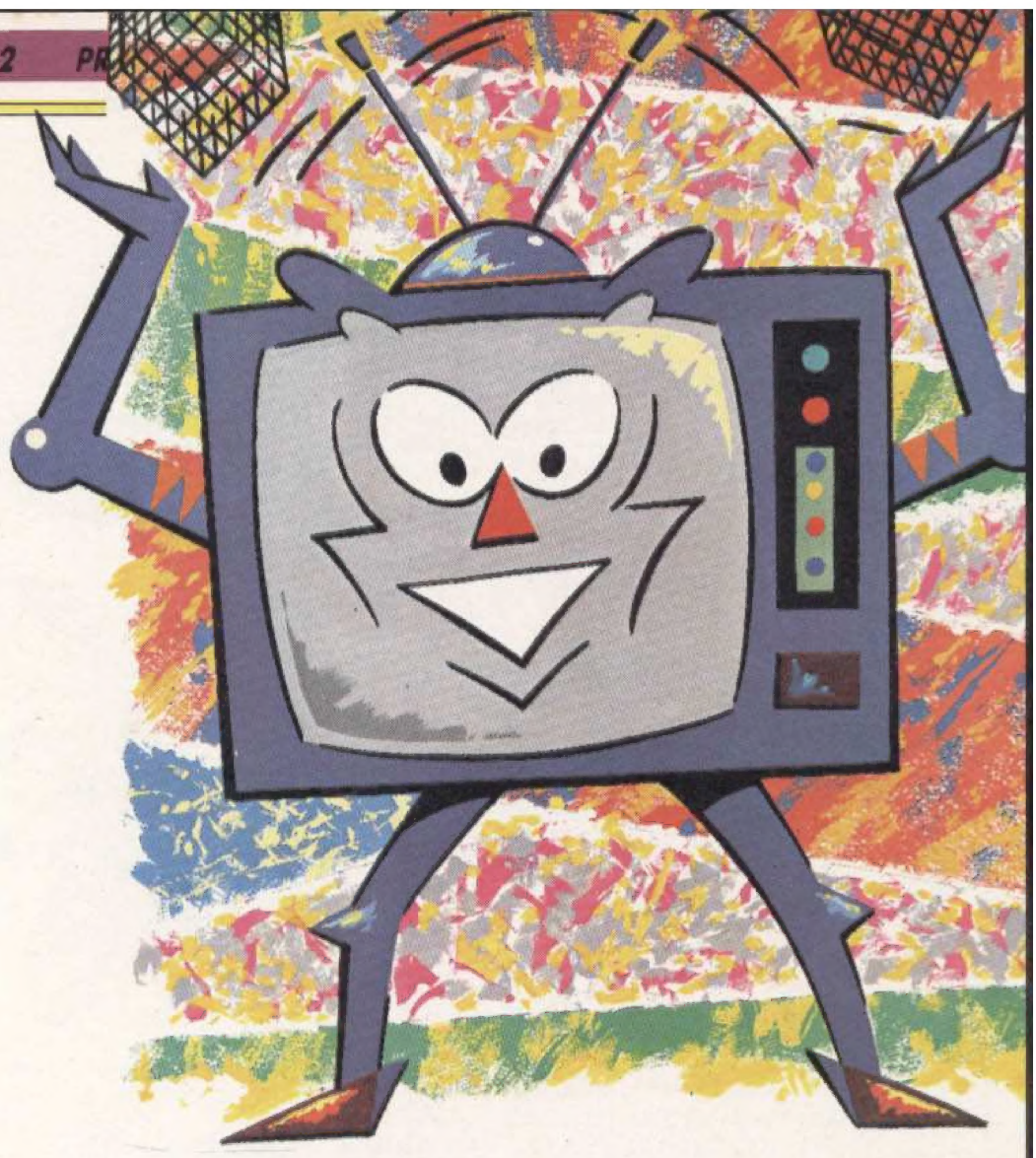
```
100 PI = 4 *  ATN (1)
110  GOSUB 9000
120 L = 20:N = 5
125  GOSUB 505: GOTO 140
130  GOSUB 500
140  IF X = 0 AND Y = 0 AND Z =
 0 THEN 170
150  GOSUB 1000
160  GOTO 130
170  HOME
180  TEXT : END
500  GET T$: IF T$ = "" THEN 50
0
505  TEXT : HOME
510  INPUT "POSICAO DO OBSERVAD
OR (X,Y,Z) ";X,Y,Z
520  GOSUB 8000
530  HGR2 : HCOLOR= 3
540  RETURN
1000 P = L / 2
1010 T1 = 1:T2 = 0:T3 = 0
1020 T4 = 0:T5 = 1:T6 = 0
1030 T7 =  - P:T8 =  - P:T9 =
- P
1040  GOSUB 1200: REM  BAIXO
1050 T7 =  - P:T8 =  - P:T9 = P

1060  GOSUB 1200: REM  CIMA
1070 T4 = 0:T5 = 0:T6 =  - 1
1080  GOSUB 1200: REM  ESQUERDA

1090 T7 =  - P:T8 = P:T9 = P
1100  GOSUB 1200: REM  DIREITA
1110 T1 = 0:T2 =  - 1:T3 = 0
1120  GOSUB 1200: REM  TRAS
1130 T7 = P:T8 = P:T9 = P
1140  GOSUB 1200 REM  FRENTE
1170  RETURN
1200 XA = 0:YA = 0:LW = L:LH =
L:NX = N:NY = N
1210  GOSUB 5000
1220  RETURN
```

```
 110 GOSUB 9000
 120 LET L=20: LET N=3
 125 GOSUB 505: GOTO 140
 130 GOSUB 500
 140 IF X=0 AND Y=0 AND Z=0
THEN  GOTO 170
 150 GOSUB 1000
 160 GOTO 130
 170 CLS
 180 STOP
 500 IF INKEY$=""  THEN   GOTO
  500
 505 CLS
 510 INPUT "INTRODUZA POSICAO D
O OBSERVADOR (X,Y,Z)",X,Y,Z
 520 GOSUB 8000
 530 RETURN
1000 LET P=L/2
1010 LET T1=1: LET T2=0: LET T3
=0
1020 LET T4=0: LET T5=1: LET T6
=0
1030 LET T7=-P: LET T8=-P: LET
T9=-P
```

```
1040 GOSUB 1200: REM BAIXO
1050 LET T7=-P: LET T8=-P: LET
T9=P
1060 GOSUB 1200: REM CIMA
1070 LET T4=0: LET T5=0: LET T6
=-1
1080 GOSUB 1200: REM ESQUERDA
1090 LET T7=-P: LET T8=P: LET T
9=P
1100 GOSUB 1200: REM DIREITA
1110 LET T1=0: LET T2=-1: LET T
3=0
1120 GOSUB 1200: REM TRAS
1130 LET T7=P: LET T8=P: LET T9
=P
1140 GOSUB 1200: REM FRENTE
1170 RETURN
1200 LET XA=0: LET YA=0: LET LW
=L: LET LH=L: LET NX=N: LET NY=
N
1210 GOSUB 5000
1220 RETURN
```



```
100 PI=4*ATN(1):PMODE 4,1
110 GOSUB 9000
120 L=20:N=5
125 GOSUB 505:GOTO 140
130 GOSUB 500
140 IF X=0 AND Y=0 AND Z=0 THEN
 170
150 GOSUB 1000
160 GOTO 130
170 CLS
180 END
500 IF INKEY$="" THEN 500
505 CLS
510 INPUT" INTRODUZA A POSICAO
DO OBSERVA- DOR (X,Y,Z) ";X,Y,Z
520 GOSUB 8000
530 PCLS:SCREEN 1,1
540 RETURN
1000 P=L/2
1010 T1=1:T2=0:T3=0
1020 T4=0:T5=1:T6=0
1030 T7=-P:T8=-P:T9=-P
1040 GOSUB 1200 'BAIXO
1050 T7=-P:T8=-P:T9=P
1060 GOSUB 1200 'CIMA
1070 T4=0:T5=0:T6=-1
1080 GOSUB 1200 'ESQUERDA
1090 T7=-P:T8=P:T9=P
1100 GOSUB 1200 'DIREITA
1110 T1=0:T2=-1:T3=0
1120 GOSUB 1200 'ATRAS
1130 T7=P:T8=P:T9=P
1140 GOSUB 1200 'FRENTE
1170 RETURN
1200 XA=0:YA=0:LW=L:LH=L:NX=N:N
Y=N
1210 GOSUB 5000
1220 RETURN
```

Agora, rode o programa. Se ele estiver funcionando corretamente, o computador acessará a rotina de inicialização que começa na linha 110. Essa rotina inicializa algumas variáveis e imprime uma mensagem que pede um valor para **D**, a distância do plano de projeção. A perspectiva nesse programa é feita de tal maneira que, quanto maior a distância, mais afastado aparecerá o objeto. Devemos entrar com o valor 1000 para começar. O programa retorna para a linha 120, que especifica o comprimento (**L**) de cada lado do cubo, e o número de grades quadradas (**N**) ao longo de cada lado. As linhas 130 a 160 lêem as coordenadas da posição do olho (linha 510), e desenham o cubo a partir dessa posição. Assim que obtivermos a primeira imagem do objeto desenhado, podemos digitar um novo conjunto de coordenadas e ver o cubo por um outro ângulo. Quando entramos os valores 0, 0 e 0, o programa termina. A rotina das linhas 500 a 530 chama a rotina de posição (linha 120) para inicializar as constantes de transformação. A rotina do Cubo (linhas 1000 a 1170) posiciona e desenha cada um dos seis lados. A rotina Lado (linhas 1200 a 1220) desenha cada lado como uma grade, usando a Rotina da Grade.

Tente valores diferentes para a posição do olho e estude os efeitos decorrentes. O valor 1000 para **D** e 250, 0 e 0 para X, Y e Z são bons para começar. Depois, adote 100 para **D** e 25,0 e 0 para X, Y e Z. O cubo deverá aparecer do mesmo tamanho, mas a perspectiva será muito mais pronunciada. Para digitar novos valores para X, Y e Z devemos pressionar qualquer tecla e a mensagem aparecerá. Para atribuir novos valores a **D**, interrompa o programa com <**BREAK**> ou <**RESET**> e rode-o novamente. Uma mensagem aparecerá para **D**, seguida de outra para X, Y e Z.

Como o programa ignora eventuais erros ocasionados por pontos fora da tela, os resultados podem ser bastante estranhos (neste caso, uma linha inteira é omitida, mesmo quando apenas uma de suas extremidades excede o ponto máximo permitido). Para pontos mais próximos do que uma certa distância mínima e pontos localizados atrás do olho, ZN será ignorado. Dessa maneira, se a posição do olho estiver muito próxima, ou mesmo dentro do cubo, poderão ocorrer falsos resultados.

Finalmente, é aconselhável guardar uma cópia da listagem completa em fita ou disco, de modo a usar as mesmas rotinas para desenhar formas duplicadas e produzir gráficos circulares em futuros programas.

CONECTE UMA IMPRESSORA

Ao contrário das máquinas de escrever, sempre prontas para serem operadas, as impressoras só podem ser usadas com um computador depois que certas condições são satisfeitas.

Se você leu *Como Escolher uma Impressora* (página 521), já sabe como agir para adquirir um desses periféricos.

Dependendo da marca e do modelo do micro, além da impressora, você precisará provavelmente de uma interface que a compatibilize com ele.

Mas adquirir uma impressora e sua respectiva interface pode não resolver todos os problemas. Com efeito, muitos usuários descobrem, após a compra, que ainda precisam de um bom tempo para ligar o novo periférico e fazê-lo funcionar de modo fluente. É que isso exige informações complementares sobre quais são os códigos de controle adequados, como enviá-los à impressora e como trabalhar com todas aquelas funções especiais, que pareciam prometer tanto e que o fizeram sair de casa e adquirir a impressora.

Neste artigo, tentaremos esclarecer alguns dos pontos ainda obscuros e orientá-lo no sentido de resolver as dificuldades mais comuns que surgem para aqueles que desejam se equipar com uma dessas máquinas.

COMO AJUSTAR A IMPRESSORA

Suponhamos que você já adquiriu uma impressora, que a conectou à tomada, e que está agora no ponto em que não sabe como fazer para colocar tudo aquilo em funcionamento...

O primeiro conselho é: não ligue ainda o periférico. Existem uma ou duas coisas que você deve fazer primeiro; para muitas impressoras, o mais importante de tudo é a remoção do parafuso de fixação para transporte (usado para imobilizar a máquina quando ela é levada de um lugar para outro).

Algumas impressoras não têm mecanismo de proteção para transporte; portanto, verifique com bastante atenção o manual de instruções do fabricante. Geralmente, existem dois ou três parafusos, que podem ser auxiliados por elásticos ou fitas adesivas. Estes devem ser removidos cuidadosamente, pois as últimas coisas de que você vai necessitar quando estiver imprimindo serão elásticos e fitas adesivas. Quanto aos parafusos, mantenha-os à mão: você talvez precise reutilizá-los em caso de transporte da impressora.

- COMO AJUSTAR A IMPRESSORA
- QUE PAPEL USAR
- TESTE A IMPRESSORA
- A CONEXÃO COM O COMPUTADOR
- COMO FAZER O CONTROLE
- AS OPÇÕES DE HARDWARE
- EFEITOS ESPECIAIS
- CÓDIGOS DE ESCAPE
- LIGUE O ZX-81 E O SPECTRUM
- IMPRESSORAS DO FUTURO

Uma vez destravado o aparelho, você pode ligá-lo. O botão para isso geralmente está em uma das laterais ou na parte traseira da máquina.

Muitas impressoras dispõem de uma pequena luz, vermelha ou verde, destinada a avisar se a fonte está ligada. Mas, ainda que ela não seja provida de luzes, você provavelmente perceberá o pequeno movimento que a cabeça de impressão realiza para ajustar-se. Certos modelos avisam quando o papel não está colocado e ajustado corretamente, por meio da emissão de um alarma sonoro ou outro sinal qualquer.

Se a impressora não funcionar na primeira tentativa, não se desespere: pode ser alguma coisa tão simples como um fio solto. Se, ao contrário, ela funcionar (o que é, aliás, muito mais provável), desligue-a novamente e coloque o papel (e a fita impressora, caso isto seja necessário).

O PAPEL

Há dois tipos básicos de impressora quanto ao tipo de papel usado: as que empregam formulários contínuos (papel dobrado em z, com perfurações nas margens) e as que utilizam folhas soltas ou papel alimentado por bobinas. As máquinas mais versáteis — como as compatíveis com a linha Epson — aceitam os dois tipos de papel. Impressoras térmicas e eletrostáticas requerem papel especial e em geral são máquinas específicas para determinadas marcas de computador.

Se você puder escolher, não use papel muito fino ou excessivamente grosso. E, quando for ajustá-lo na impressora, verifique primeiro se ele não está frouxo ou demasiado tenso, pois o ajuste incorreto pode provocar rompimentos no papel ou levá-lo a enredar-se no mecanismo.

AUTOTESTE

No caso de sua impressora dispor de recursos para um autoteste, é conveniente acioná-los assim que ela estiver pronta para trabalhar. Esse tipo de verificação consiste em imprimir ininterruptamente um conjunto completo de caracteres. Ele serve para dois propósitos: primeiro, mostrar que a impressora está funcionando independentemente do computador; segundo, para exibir o conjunto de caracteres disponíveis no periférico.

Nas impressoras de tipo mais comum, o autoteste é acionado quando se pressiona o botão de avanço de linha ou de formulário.

Para parar o autoteste, basta desligar a impressora. (Algumas máquinas dispõem de um botão especial para acionar o autoteste.)

FITAS DE IMPRESSÃO

A menos que a sua impressora seja térmica ou eletrostática, ela provavelmente precisará de uma fita de impressão. Essas fitas costumam ser vendidas em cartuchos descartáveis, e não é difícil trocá-las (consulte antes o diagrama existente no manual da impressora). Um pequeno botão na parte de cima do cartucho da fita serve para dar a ela a tensão adequada.

A CONEXÃO AO COMPUTADOR

A última e mais importante etapa dessa seqüência é a ligação da impressora ao computador. Em muitos casos isso é tão simples quanto inserir o cabo de conexão no soquete correto: essa é a forma de ligação das impressoras que usam o padrão Centronics (paralelo). Freqüentemente, como acontece com o TK-2000 e com o MSX, existe um conector claramente reservado para a impressora. Nos computadores da linha Sinclair (ZX-81 e Spectrum), é necessário uma interface especial, que é conectada no mesmo ponto destinado a ligar outros tipos de expansão (memória, sintetizadores de voz etc.). Nesse caso, é preciso, quase sempre, desligar uma expansão já existente, para poder colocar a interface da impressora, a não ser que

você tenha comprado uma interface com um conector adicional de expansão. Esse sistema é conhecido como "encadeamento em margarida" e deve ser feito com certa precaução.

Se você vai adquirir uma impressora que não é específica para o seu computador, deve tomar alguns cuidados. Se o seu micro é um MSX, ou um TK-2000, por exemplo, a impressora deve ser, obrigatoriamente, do tipo paralelo, padrão Centronics (a maioria das impressoras mais populares, como a Mônica, a Grafix e outras compatíveis com os modelos Epson e Seikosha tem esse padrão). Os micros da linha Apple podem trabalhar tanto com impressoras paralelas como com as seriais, que usam o padrão RS-232C, dependendo da interface que se vai utilizar. Já os micros compatíveis com o TRS-Color usam apenas impressoras seriais.

Ao conectar o cabo, você notará que a maioria das impressoras tem soquetes que impedem uma ligação em posição invertida. Isso não ocorre com a impressora para os micros da linha TRS-80 (por exemplo, o CP-500), que aceitam o soquete em qualquer posição.

Se você quiser conectar uma impressora de fabricante independente nos micros da linha Sinclair (Spectrum e ZX-81), precisará adquirir uma interface especial. As interfaces para tais micros existentes no mercado nacional devem ser ligadas a impressoras paralelas com o padrão Centronics.

Em alguns casos, a interface só funciona quando o computador é carregado com um programa específico para isso. Entretanto, uma vez que você tenha uma, poderá ligar praticamente qualquer impressora ao seu computador. As melhores interfaces são oferecidas pelos próprios fabricantes de micros, mas mesmo assim é bom fazê-las funcionar antes de adquiri-las.

COMANDOS DE CONTROLE

Qualquer coisa a ser impressa tem que ser enviada primeiro pelo computador à impressora, através do cabo de conexão. Os comandos usados para controlar a impressora variam de computador para computador, e em geral consistem de algumas instruções simples.

SS

O Spectrum e o ZX-81 se comunicam com a impressora por comandos simples em BASIC, de fácil manipulação. O comando **LPRINT** tem um efeito similar ao **PRINT**; com ele, porém, a saída vai para a impressora e não para a tela. No ZX-81, todas as regras do **PRINT** valem também para o **LPRINT**. No Spectrum, entretanto, não se tem a mesma liberdade de especificar o tipo de impressão, como se faz com o comando **PRINT**. Por exemplo, não é possível utilizar os comandos **PAPER, INK, OVER, FLASH** e **BRIGHT**. O **INVERSE** só funciona em certos tipos de impressoras gráficas. O mesmo acontece com os caracteres gráficos que podem ser colocados dentro de um **PRINT**, tanto no ZX-81 quanto no Spectrum (as impressoras não específicas para tais micros não dispõem desses caracteres no seu conjunto de impressão). O **PRINT AT** também não funciona na impressora.

Assim como o comando **PRINT** é substituído por **LPRINT**, o **LIST** dá lugar a **LLIST**, que lista o programa da memória na impressora. Quando a listagem ocupar mais de uma tela, **LLIST** gerará uma impressão contínua, independentemente do tamanho do programa.

O terceiro comando BASIC usado para acionar a impressora é o **COPY**. Ele copia o que estiver na tela por um processo denominado *descarregamento de tela*. Quando a impressora é utilizada para produzir pontos em vez de letras é chamada de impressora gráfica. Você pode trabalhar com o **COPY** para fazer uma cópia da tela contendo os resultados de um texto ou de uma listagem de programa em qualquer tipo de impressora. A cópia de telas gráficas funcionará apenas se a impressora for específica para o seu computador (uma impressora gráfica, portanto).

O TK-90X emprega uma interface paralela, própria para acionar uma impressora e que exige o carregamento prévio de um programa de controle *(driver)*. Esse programa é vendido em conjunto com a interface, com um cabo de conexão e com um manual de instruções. O software só precisa ser carregado uma vez ao se ligar o computador. Mas, sempre que este for desligado, será preciso carregá-lo novamente. O software permite a utilização dos comandos **LPRINT, LLIST** e **COPY**.

Já os micros da linha ZX-81, como o TK-85, requerem também uma interface paralela; mas não é preciso carregar um software de controle para usar diretamente os comandos citados.

Os micros da linha Apple podem ser conectados tanto a impressoras seriais como a impressoras paralelas. Para isso, é necessário adquirir separadamente a interface adequada de controle; esta deve ser encaixada em um dos soquetes internos do console do computador. De forma geral, costuma-se dar preferência à interface paralela, uma vez que é maior o número de impressoras que seguem esse padrão.

A impressora é tratada, pelo BASIC e pelo sistema operacional, como um dispositivo de saída que recebe o número lógico correspondente ao soquete no qual foi inserida a placa de controle. Assim, se encaixarmos a placa no soquete 1 (a localização mais comum) e quisermos ativar a impressora, é necessário digitar o comando em BASIC:

```
PR#1
```

A partir desse momento, todo **PRINT** ou **LIST** que for digitado em modo direto, ou encontrado dentro de um programa, terá sua saída direcionada tanto para a tela quanto para a impressora (modo eco). Portanto, o **PR** pode ser colocado em pontos selecionados de um programa, para "ligar" e "desligar" o eco na impressora.

Os micros da linha Apple não dispõem, como os de outras linhas, de comandos específicos para a impressora, como **LLIST, LPRINT** etc. Por motivos óbvios, os comandos de posicionamento do cursor, como **HTAB** e **VTAB** não funcionam com a impressora.

MSX

A interface de comunicação adequada aos micros da linha MSX é do tipo paralela padrão Centronics. Tanto ela quanto o conector para o cabo de ligação já estão incluídos na configuração básica da máquina.

Existem três protocolos de impressão (conjuntos de caracteres de impressão diferentes) para micros da linha MSX. O primeiro padrão é o MSX Internacional, que segue a tabela de caracteres ASCII especiais e gráficos do padrão MSX original. Esse padrão é seguido por mais de trinta fabricantes de todo o mundo. O segundo é o ABICOMP (Associação Brasileira de Indústrias de Computadores): ele segue normas nacionais, que incluem os caracteres acentuados da língua portuguesa. Finalmente, temos um terceiro protocolo, criado no Brasil pela Gradiente para os primeiros modelos do Expert MSX. Ao se adquirir uma impressora, é necessário especificar qual o padrão seguido pelo micro; caso contrário, a impressora não funcionará corretamente.

Os comandos usados com o BASIC

para o controle da impressora são também bastante simples e diretos.

O comando **LPRINT** tem um efeito similar ao **PRINT**; com ele, porém, a saída vai para a impressora e não para a tela. Todas as regras de sintaxe que valem para o **PRINT** servem para o **LPRINT**, inclusive quando se trata do emprego de **TAB** e **USING**. Entretanto, os comandos de especificação de cores (como o **COLOR**) e os de localização bidimensional do cursor (como o **LOCATE**) não funcionam na impressora. O mesmo acontece com os caracteres gráficos e especiais que podem ser colocados dentro de um **PRINT**, uma vez que impressoras não específicas para esse micro não dispõem desses caracteres no seu conjunto de impressão.

Assim como o comando **PRINT** é substituído por **LPRINT**, o comando **LIST** é substituído por **LLIST**, que lista o programa da memória na impressora. Quando a listagem tomar mais de uma tela, **LLIST** gerará uma impressão contínua independentemente do tamanho do programa. As regras de utilização do **LIST** valem para o **LLIST**. Outra função útil, de uso exclusivo da impressora, é a **LPOS**, que informa ao programa a posição lógica da cabeça de impressão, em número de colunas a partir da margem esquerda de uma linha.

Também nos micros da linha TRS-80 a interface de comunicação é do tipo paralela padrão Centronics, e ela e o conector para o cabo de ligação incluem-se na configuração básica das máquinas mais completas, como o CP-500.

Os comandos usados com o BASIC para o controle da impressora são simples e diretos. O **LPRINT** tem um efeito similar ao **PRINT**, embora direcione a saída para a impressora e não para a tela. Todas as regras de sintaxe do **PRINT** valem para o **LPRINT** (o que inclui as expressões **TAB** e **USING**). Entretanto, os comandos de localização bidimensional do cursor como **PRINT@** não funcionam com a impressora. O mesmo se dá com os caracteres gráficos e especiais que podem ser colocados dentro de um **PRINT**, já que impressoras não específicas para o TRS-80 não têm esses sinais no seu conjunto de impressão.

Assim como o comando **PRINT** dá lugar a **LPRINT**, o **LIST** é substituído por **LLIST**, que lista o programa da memória na impressora. Se a listagem tomar mais de uma tela, **LLIST** gerará uma impressão contínua independentemente do tamanho do programa. As regras do **LIST** valem para o **LLIST**.

Outra função útil, de uso exclusivo da impressora, é a **LPOS**, que informa o programa a respeito da posição lógica da cabeça de impressão, em número de colunas a partir da margem esquerda.

Os micros da linha TRS-Color dispõem só de uma porta serial para comunicação com periféricos, que é a usada para conexão com a impressora. Mesmo assim, o controle da impressão (feita por alguns comandos do BASIC) nesses micros também é simples. Para listar um programa digite **LLIST** (seguido por *<ENTER>*). Todas as regras do **LIST** valem para o **LLIST**, só que a saída vai para a impressora, e não para a tela. Para usar **PRINT** com a impressora, é preciso digitar a expressão:

```
PRINT#-2,"TEXTO"
```

Ela determina ao computador que envie sua saída (aqui a palavra **TEXTO**) ao periférico **-2**, que é o número de identificação para a impressora (mais exatamente, para a interface serial). Repare que há uma vírgula após o **-2** no comando acima. Ela informa ao computador que a seguir vem um texto para ser impresso. Além disso, se usado com impressoras capazes de imprimir letras minúsculas, o texto que está em vídeo inverso na tela aparecerá em minúsculas na impressora (é necessário pressionar **<SHIFT><0>** antes).

Assim, o comando **PRINT#-2** tem um efeito similar ao **PRINT** (com uma diferença: ele determina que a saída vá para a impressora, e não para a tela). As regras de sintaxe do **PRINT** valem para o **PRINT#-2**, inclusive quando aparecem as expressões **TAB** e **USING**. Porém, os comandos de especificação de cores, e de localização bidimensional do cursor, como **PRINT@**, não funcionam com a impressora. O mesmo se dá com os caracteres gráficos e especiais que podem ser colocados dentro de um

PRINT, pois as impressoras não específicas para o TRS-Color não têm esses sinais em seu conjunto de impressão.

Outra função de uso exclusivo da impressora é a **POST(-2)**: informa a posição lógica da cabeça de impressão, em número de colunas a partir da margem esquerda de uma linha.

PROGRAMAÇÃO INTERNA DA IMPRESSORA

Quem deseja dominar os segredos do trabalho com impressoras precisa conhecer primeiro como se faz sua programação interna e externa. A programação interna é feita normalmente por meio de microinterruptores (os *DIP*) e fios removíveis de conexão (os *estrapes*), situados na placa interna da máquina. Já a externa é realizada através de seqüências de caracteres de controle (denominados *seqüências de escape* por começarem com o código ASCII 27, correspondendo ao caractere de controle chamado ESCAPE).

Dentro das impressoras e de outros equipamentos eletrônicos, você encontrará um conjunto de pequenas chaves: as chaves DIP. Elas podem ser usadas para alterar o estado da impressora, quando ela for ligada (o estado *default*, ou predefinido). São comumente inicializados por chave o comprimento da linha, o espacejamento, o retorno do carro e o comprimento da página. As impressoras de tecnologia japonesa permitem ainda que você selecione o conjunto de caracteres internacionais de que precisa (exemplo: o sinal da libra esterlina, ou acentos para certos caracteres, empregados em idiomas como francês, sueco e português).

Para encontrar os interruptores de programação interna, consulte o manual da impressora: em geral, eles estão localizados sob a tampa da impressora (esta, porém, só deve ser removida quando a impressora estiver desconectada da corrente elétrica).

Uma vez localizadas as chaves DIP, você pode trocá-las de posição empurrando-as delicadamente com os dedos ou com uma chave de fenda. O manual de instruções deve ter um diagrama do DIP e de suas posições, de modo a mostrar para que elas estão selecionadas. Faça um teste você mesmo.

Outras características de uma impressora (como, por exemplo, o fato de todo caractere de alimentação de linha ser seguido automaticamente de um retorno de carro) podem ser alteradas removendo-se ou inserindo-se um estrape (pequeno fio em forma de U, com conectores em ambas as pontas) em pontos apropriados da placa eletrônica da máquina. Antes de fazer isso, porém, é conveniente consultar o manual específico da sua impressora.

CÓDIGOS DE ESCAPE

Algumas impressoras (como a Mônica, por exemplo) admitem o envio de códigos de controle, da mesma forma que se enviam os dados a serem impressos. Esses códigos são freqüentemente chamados de seqüências de escape. Assim, muitas impressoras esperam que você coloque o código para o escape, **CHR$(27)**, antes do código de controle. Isso permite à impressora saber que o que está chegando no momento é uma mensagem de programação externa, e não um caractere a ser impresso.

Cada impressora tem as suas seqüências de escape próprias, embora os caracteres usados nas máquinas compatíveis com a Epson estejam se tornando rapidamente um padrão para impressoras que oferecem facilidades similares. Nas máquinas mais versáteis há um enorme conjunto de seqüências de escape. Estas podem ser enviadas por um comando **LPRINT** normal.

Em uma Epson, o exemplo a seguir imprime a palavra COMPUTADOR em modo enfatizado, desde que tenham sido enviados os comandos corretos.

```
LPRINT CHR$(27);'E';'COMPUTADOR
'
```

```
PR#1
PRINT CHR$(27);'E';'COMPUTADOR'
```

```
PRINT#-2,CHR$(27);'E';'COMPUTAD
OR'
```

Mas não se esqueça de voltar ao modo normal. Para isso, use os seguintes comandos:

```
LPRINT CHR$(27);'F'
```

```
PR#1
PRINT CHR$(27);'F'
```

```
PRINT#-2,CHR$(27);'F'
```

Lembre-se, porém, de que esses códigos são exclusivamente para as impressoras Epson e as que são compatíveis com ela. Para outras impressoras, os códigos serão diferentes.

OS CÓDIGOS EPSON

Os efeitos mais comuns obtidos com os códigos de escape do tipo Epson são os seguintes: impressão em itálico, impressão grifada, impressão em negrito, subscritos e sobrescritos. Certas operações variam de máquina para máquina. As que apresentamos a seguir funcionam na maioria das impressoras compatíveis com a Epson:

ESC 4 ativa um conjunto alternativo de caracteres (geralmente o itálico).
ESC 5 retorna ao conjunto padrão.
ESC - ativa e desativa o grifo. Deve ser seguido por **CHR$(1)** ou por **CHR$(49)**; para a alteração, é desativado por **CHR$(0)** ou **CHR$(48)**, conforme tenha sido ativado por **CHR$(1)** ou por **CHR$(49)**.
ESC G ativa o modo dupla impressão.
ESC H desativa o modo dupla impressão.
ESC S, quando é seguido de **CHR$(0)**, ativa o modo de sobrescrição; quando é seguido de **CHR$(1)**, ativa o modo de subscrição.
ESC T retorna a impressora ao modo normal depois de ativado o modo de subscrição ou sobrescrição.
ESC @ inicia a impressora e volta todos comandos aos valores originais.

Você pode usar as seqüências de escape em praticamente todas as impressoras conhecidas, colocando-as depois do comando que o seu computador usa para enviar uma saída à impressora (**LPRINT**, **PRINT#-2**, ou qualquer outro) e separando-as por pontos e vírgulas, assim como se faria com qualquer outro comando **CHR$**.

Algumas impressoras exigem que se coloque um caractere **ASCII** para o ESCAPE entre todos os códigos de controle. O manual de instruções deve dizer se você vai ou não precisar disso.

NOVAS TENDÊNCIAS

As seqüências de escape terão um papel mais importante na medida em que as impressoras se tornarem mais baratas e sofisticadas. Já existem impressoras com matrizes de ponto cuja qualidade é quase a mesma das impressoras "margarida"; algumas delas têm características próprias usualmente associadas com fotocomposição profissional, tais como: espacejamento proporcional e justificação automática à direita ou à esquerda.

FELIZ ATERRISSAGEM

- UMA ROTINA DE CONTROLE DO AVIÃO PELO TECLADO
- A ATERRISSAGEM
- ACIDENTES AÉREOS
- PERDAS E DANOS

Agora que você já é um (relativo) ás da aviação, desligue o piloto automático e, com apenas três funções, faça o aparelho aterrissar. Mas tenha cuidado com os urubus.

Última parte de nosso programa simulador de vôo, este artigo fará com que você possa finalmente assumir o controle direto do avião. Três pares de teclas permitirão aumentar e diminuir a rotação do motor, virar o aparelho para os lados, subir ou descer.

Nossa viagem começou a 20.000 metros do centro da pista e a 2.000 metros de altura.

Pilotar pode ser uma tarefa difícil... sobretudo se você nada sabe do assunto. Não espere bons resultados nos primeiros vôos: como em tudo na vida, a habilidade só vem com o tempo.

Relativamente lentos por estarem em BASIC, programas como esse parecem ainda mais morosos pelo fato de não permitirem mudanças bruscas de curso. Com paciência e perseverança, porém, você conseguirá controlar o aparelho e melhorar seus pousos. O programa o manterá informado a respeito de cada aterrissagem ou acidente. Essas mensagens são importantes: será por meio delas que você saberá não só o que está acontecendo em cada momento como também onde se encontram suas principais deficiências.

```
3000 LET POW=0: LET KS=INKEYS:
IF KS="" THEN  RETURN
3010 IF KS="S" THEN  LET POW=-1
3020 IF KS="F" THEN  LET POW=1
3030 IF KS="Q" THEN  LET PT=PT+
1
3040 IF KS="A" THEN  LET PT=PT-
1
3050 IF KS="O" AND RL>-30 THEN
 LET RL=RL-1
3060 IF KS="P" AND RL<30 THEN
LET RL=RL+1
```

```
3070 RETURN
5020 CLS : INPUT "VELOCIDADE DO
 VENTO (1-50 M/S)",X0
5025 IF X0>50 OR X0<1 THEN  GOT
O 5020
5030 INPUT "DIRECAO DO VENTO (0
-359 GRAUS)",X1
5035 IF X1>359 OR X1<0 THEN  GO
TO 5030
5040 LET X0=X0/3
5050 PRINT '"VELOCIDADE DO VENT
O= ";3*X0;" M/S": PRINT "DIRECA
O= ";X1;" GRAUS"
5055 PAUSE 100: CLS
5060 LET WY=-X0*COS (X1*C)
5070 LET WX=-X0*SIN (X1*C)
5500 GOSUB 3000: GOSUB 1000
5510 IF PZ<=0 THEN  GOTO 6000
5520 GOSUB 2000
5530 GOTO 5500
6000 IF ABS RL>RT OR PT>TP OR P
T<0 OR VV>80 THEN  GOTO 6030
6010 IF ABS PX>WR OR ABS PY>100
0 THEN  GOTO 6060
6020 CLS : PRINT " PARABENS!":
PRINT "ATERRISSAGEM BEM SUCEDID
A.": GOTO 6100
6030 FOR N=0 TO 20 STEP .5: PLO
T 127,130: DRAW 120-INT (RND*24
0),45-INT (RND*90): SOUND .005,
20-N: NEXT N
6040 PAUSE 50
6050 CLS : PRINT "LAMENTAMOS IN
FORMAR QUE O VOO   PJ-26 SOFREU
 UM GRAVE ACIDENTE. NAO HA SOBR
EVIVENTES.": GOTO 6100
6060 CLS : PRINT "VOCE POUSOU F
ORA DA PISTA"
6070 IF VV<40 THEN  PRINT "COMO
 SUA VELOCIDADE ERA BAIXA   QUA
SE NADA ACONTECEU.": GOTO 6100
6080 IF VV<80 THEN  PRINT "ENTR
E MORTOS E FERIDOS          SAL
VARAM-SE TODOS.": GOTO 6100
6090 PRINT "NINGUEM SOBREVIVE A
 UM POUSO    NESSA VELOCIDADE!"
6100 PRINT '"VALORES RELATIVOS
AO POUSO"
6110 PRINT '"VELOCIDADE DO AR=
";INT VV;" M/S": PRINT "DISTANC
IA= ";INT (SQR (PY*PY+PX*PX)):
PRINT "INCLINACAO FRONTAL= ";PT
6120 PRINT "INCLINACAO LATERAL=
 ";RL: PRINT "ROTACAO DO MOTOR=
 ";INT (10*TC)/10;" X 1000"
6130 PRINT "DESVIO DO EIXO DA P
ISTA= ";INT ABS PX;" METROS": P
RINT "DIRECAO= ";AD;" GRAUS"
6140 PRINT '"QUER VOAR NOVAMENT
E (S/N) ?"
6150 LET A$=INKEY$: IF A$<>"S"
AND A$<>"N" THEN  GOTO 6150
6160 IF A$="N" THEN  CLS : STOP
6170 RUN
```

A sub-rotina que vai de 3000 a 3070 nos dará o controle do aparelho.

Usando o comando **INKEY$** para detectar teclas, podemos modificar a rotação do motor, virar o aparelho, subir ou descer. Uma nova variável — **POW** — aparece nas linhas 3010 e 3020; ela serve para alterar a rotação. Um valor zero é atribuído a **POW** na linha 3000 cada vez que a sub-rotina é chamada. A tecla S determina a velocidade, enquanto F acelera o motor.

Q e A são usadas para provocar a ascensão e a descida do avião, somando ou subtraindo 1 à variável **PT** nas linhas 3030 e 3040. Finalmente, podemos usar O ou P para virar o aparelho — as linhas 3050 e 3060 cuidam disto, alterando o valor de **RL**.

As linhas 5020 a 5070 permitem variar o nível de dificuldade do jogo, escolhendo a força e a direção do vento. A opção mais fácil é a de um vento de um metro por segundo em uma direção de zero grau. A partir da nossa opção, a linha 5060 calcula a velocidade do vento na direção frontal do aparelho, e a linha 5070 calcula a componente lateral dessa velocidade. **SIN** e **COS** servem para calcular as componentes em cada direção. **WX** e **WY** são usadas para alterar a posição do aparelho — GX e GY na linha 5080.

A "alma" do programa é constituída pelas linhas 5500 a 5530, que chamam todas as grandes sub-rotinas em seqüência; desse modo, a posição é modificada continuamente.

A linha 5510 verifica, por intermédio do valor de **PZ**, se o avião aterrissou. O programa vai então para o segmento que verifica se a aterrissagem foi satisfatória ou desastrosa.

A linha 6000 verifica se as inclinações lateral e frontal do avião estão fora de limites aceitáveis, ou se a velocidade é maior que oitenta metros por segundo. Se houver uma dessas condições, ocorrerá um acidente. O motivo pode ser uma asa muito baixa, excesso de velocidade ou simplesmente uma tentativa de aterrissagem fora da pista. O programa prosseguirá na linha 6030, que cuida dos acidentes. Ela traça várias linhas irradiando do centro do pára-brisa. À medida que essas rachaduras são desenhadas, o computador emite um som. Após o acidente, a linha 6040 provoca uma pausa antes que o piloto seja informado das más notícias pela linha 6050, que fornece também os detalhes do acidente.

Se o avião pousar corretamente, a linha 6010 verificará a posição da aterrissagem. Se pousar fora da pista, a linha 6060 informará sobre o fato. Se a velocidade era inferior a cinqüenta metros por segundo, a linha 6070 dirá ao piloto que houve um pouso bem-sucedido. As linhas 6080 e 6090 informarão se o acidente foi pequeno, apenas com feridos leves, ou se foi um desastre mais grave; neste caso, o jeito é procurar a caixa-preta.

MSX

Apague a linha 5505. Note que algumas linhas deste programa são modificações de linhas já existentes.

```
2 DEF USR1=&H41:DEFUSR2=&H44
5 GOTO 5000
10 SCREEN 2,2:KEY OFF:COLOR 15,
4,2
15 A=USR1(0)
50 ON KEY GOSUB 3100,3200
60 KEY(1) STOP:KEY(2) STOP
110 RETURN
3000 A=STICK(0)
3010 KEY(1) ON:KEY(2) ON
3030 IF A=1 THEN PT=PT+3:GOTO 3
070
3040 IF A=5 THEN PT=PT-3:GOTO 3
070
3050 IF A=7 AND RL>-30 THEN RL=
RL-3:GOTO 3070
3060 IF A=3 AND RL<30 THEN RL=R
L+3
3070 KEY(1) STOP:KEY(2) STOP:RE
TURN
3100 T2=TC:IF TC>.2 THEN TC=TC-
.2:RETURN
3200 T2=TC:IF TC<8.8 THEN TC=TC
+.2:RETURN
3900 LINE(X,Y)-(X+50,Y+8),2,BF
5020 SCREEN 0:KEY OFF:INPUT"Vel
ocidade do vento (1-50 m/s) ";X
0
5025 IF X0>50 OR X0<0 THEN 5020
5030 INPUT"Direção do vento (0-
359 graus) ";X1
5035 IF X1>359 OR X1<0 THEN 503
0
5040 X0=X0/3
5050 PRINT:PRINT "Velocidade do
 vento: ";INT(3*X0);"m/s":PRINT
"Direção do vento :";X1;" graus
"
5060 WY=X0*COS(X1*C):WX=-X0*SIN
(X1*C)
5070 FOR I=1 TO 1000:NEXT:GOSUB
 10
5400 A=USR2(0)
5500 GOSUB 3000:GOSUB 1000
5510 IF PZ<=0 THEN 6000
5520 GOSUB 2000
5530 GOTO 5500
6000 IF ABS(RL)>RT OR PT>TP OR
PT<0 OR VV>80 THEN 6100
6010 IF ABS(PX)>WR OR ABS(PY)>1
000 THEN 6200
6020 COLOR 15,4,4:SCREEN 0
6030 PRINT "Parabéns !":PRINT "
Aterrissagem bem sucedida.":GOT
O 6500
6100 LINE (0,0)-(RND(1)*128+64,
RND(1)*96+49),15:LINE-(255,191)
,15:LINE (255,0)-(RND(1)*128+64
,RND(1)*96+49),15:LINE-(0,191),
15
6110 FOR K=1 TO 1000:NEXT
6120 COLOR 15,4,4:SCREEN 0:PRIN
T "Lamentamos informar, que o v
ôo PJ26"
6130 PRINT "sofreu um grave aci
dente."
```

```
6140 PRINT "Não há sobreviventes.":GOTO 6500
6200 COLOR 15,4,4:SCREEN 0
6210 PRINT "Você pousou fora da pista."
6220 IF VV<40 THEN PRINT "Como sua velocodade era baixa,":PRINT "quase nada aconteceu.":GOTO 6500
6230 IF VV<80 THEN PRINT "Entre mortos e feridos,":PRINT "salvaram-se todos.":GOTO 6500
6240 PRINT "Ninguém sobrevive a um pouso:"PRINT "nesta velocidade."
6500 LOCATE 5,10:PRINT "VALORES RELATIVOS AO POUSO"
6510 PRINT:PRINT "Velocidade do ar = ";INT(VV);"m/s":PRINT "Distância = ";INT(SQR(PY*PY+PX*PX));" m":PRINT "Inclinação frontal = ";PT
6520 PRINT "Inclinação lateral = ";RL:PRINT "Rotação do motor = ";INT(TC*1000); "r.p.m."
6530 PRINT "Desvio do eixo da pista = ";INT(PX);"m":PRINT "Direção = ";AD;"graus"
6540 PRINT:PRINT "Quer voar novamente (S/N) ?"
6550 A$=INKEY$:IF A$<>"S" AND A$<>"N" THEN 6550
6560 IF A$="N" THEN CLS:END
6570 RUN
```

O laço principal do programa fica nas linhas 5500 a 5530. A primeira coisa que ele faz é chamar uma sub-rotina na linha 3000 que permitirá o controle do avião. Essa sub-rotina usa o comando **STICK**, por meio do qual podemos pilotar com o teclado — ou com o joystick, se usarmos **STICK (1)**.

As linhas 3030 e 3040 controlam os movimentos de subida e descida, enquanto as linhas 3050 e 3060 determinam os movimentos laterais.

É necessário ainda que existam formas de acelerar e diminuir a rotação do motor. Isso é feito por intermédio das teclas de função do MSX. Nessa rotina serão utilizadas as duas primeiras teclas de função. Para tanto, foi incluída a linha 50 no início do programa; ela informa ao computador o que fazer quando uma das teclas de função for pressionada. A função responsável por isto é **ON KEY GOSUB**, que difere da função **ON "variável" GOSUB** em diversos aspectos. O programa não será desviado ao encontrar a linha 50; ele apenas ficará sabendo para onde ir no momento oportuno, guardando os endereços na memória.

Só quando o programa chegar à linha 3010 é que as teclas de função serão ativadas pelo comando **KEY (N) ON**, onde **N** é o número da tecla. A partir daí, se pressionarmos a tecla de função 1, o programa executará a sub-rotina da linha 3100, que desacelera o motor. Se a tecla de função pressionada for a 2, o motor será acelerado pela sub-rotina da linha 3200.

Observe que o computador interromperá o que estiver fazendo para acelerar ou desacelerar o motor, voltando em seguida para a tarefa anterior. Contudo, se o programa estiver fazendo os cálculos no momento em que pressionarmos uma das teclas de função, coisas estranhas podem acontecer. É que a primeira parte dos cálculos terá sido feita com o valor antigo da rotação do motor, e a segunda, com o valor novo.

A solução não está em desligar as teclas de função durante a execução dos cálculos, pois, neste caso, o programa só aceitaria mudanças na rotação do motor em determinados momentos.

O problema, contudo, pode ser resolvido. Para isso, é necessário que se digite o comando **KEY (N) STOP** nas linhas 60 e 3070. Depois de um comando desse tipo, o computador será capaz de detectar a tecla de função correspondente, mas o desvio não ocorrerá até que um comando do tipo **KEY (N) ON** seja encontrado. Dessa maneira, o programa "sabe" que uma tecla de função foi pressionada, e se lembrará de tal fato no momento adequado.

Algumas linhas iniciais também foram modificadas. A linha 2 define o endereço de duas rotinas em linguagem de máquina, que ficam na ROM. A primeira "desliga" temporariamente a tela, quando chamada pela linha 15, impedindo que o jogador veja o desenho que está sendo feito na tela. A segunda "liga" a tela novamente, quando chamada pela linha 5400, de forma que a cabine apareça instantaneamente.

A linha 5 desvia o programa para que o jogador informe a velocidade e a direção do vento, nas linhas que vão de 5020 a 5070. Isso é feito antes que o desenho da cabine de comando seja feito pela sub-rotina 10.

Se a linha 5510 perceber que o avião atingiu o solo, o programa prosseguirá na linha 6000. Ali serão verificados os valores das inclinações frontal e lateral, assim como as da velocidade do vento. Se todos eles estiverem dentro dos limites aceitáveis, a linha 6020 informará ao piloto que houve uma aterrissagem bem-sucedida.

Um pouso fora da pista não significa necessariamente uma tragédia. Se a velocidade do avião for inferior a 40 m/s, a linha 6210 dirá que os danos foram pequenos. Se ela estiver entre 40 e 80 m/s, a linha 6220 dará conta de que as perdas materiais foram pequenas, mas que há alguns feridos. Se a velocidade for maior que 80 m/s só nos restará procurar a caixa-preta: o desastre terá sido total.

As linhas 6500 a 6530 mostram os valores de todas as variáveis no momento do pouso. Estudando bem esses números, você aprenderá a fazer aterrissagens cada vez melhores.

Finalmente, as linhas 6540 a 6570 nos dão a opção de jogar novamente.

Antes de mais nada, apague a linha 5505. O programa a seguir completa a simulação de vôo. Use as teclas Z, X, P e L para jogar.

```
3000 K =  PEEK ( - 16384): POKE - 16368,0
3010  IF K = 193 AND TC > .2 THEN TC = TC - .2: GOTO 3070
3020  IF K = 209 AND TC < 8.8 THEN TC = TC + .2: GOTO 3070
3030  IF K = 208 THEN PT = PT + 3: GOTO 3070
3040  IF K = 204 THEN PT = PT - 3: GOTO 3070
3050  IF K = 218 AND RL >  - .005 THEN RL = RL - .001: GOTO 3070
3060  IF K = 216 AND RL < .005 THEN RL = RL + .001: GOTO 3070
3070  RETURN
5020  HOME : TEXT : INPUT "VELOCIDADE DO VENTO (1-50) M/S ";X0
5025  IF X0 > 50 OR X0 < 1 THEN 5020
5030  INPUT "DIRECAO DO VENTO (0-359) GRAUS ";X1
5035  IF X1 > 359 OR X1 < 0 THEN 5030
5040 X0 = X0 / 3
5060 WY = X0 *  COS (X1 * C)
5070 WX =  - X0 *  SIN (X1 * C)
5500  GOSUB 3000: GOSUB 1000
5510  IF PZ <  = 0 THEN 6000
5520  GOSUB 2000
5530  GOTO 5500
5570  GOTO 6550
6000  IF ( ABS (RL) > RT) OR PT > TP OR PT < 0 OR AS > 80 THEN 6100
6010  IF  ABS (PX) > WR OR  ABS (PY) > 1000 THEN 6200
6020  TEXT : HOME : PRINT "PARABENS !": PRINT "ATERRISSAGEM BEM SUCEDIDA": GOTO 6500
6100  HCOLOR= 3: HPLOT 0,0 TO RND (1) * 140 + 71, RND (1) * 96 + 49 TO 279,191: HPLOT 279,0 TO  RND (1) * 140 + 71, RND (1) * 96 + 49 TO 0,191
6110  FOR K = 1 TO 2000: NEXT
6120  TEXT : HOME : PRINT "LAMENTAMOS INFORMAR QUE O VOO PJ25" : PRINT "SOFREU UM GRAVE ACIDENTE.": PRINT "NAO HA SOBREVIVENTES.": GOTO 6500
6200  TEXT : HOME : PRINT "VOCE POUSOU FORA DA PISTA."
```

```
6210  IF AS < 40 THEN  PRINT "C
OMO SUA VELOCIDADE ERA BAIXA, "
: PRINT "NADA ACONTECEU" GOTO 6
500
6220  IF AS < 80 THEN  PRINT "E
NTRE MORTOS E FERIDOS": PRINT "
SALVARAM-SE TODOS": GOTO 6500
6230  PRINT "NINGUEM SOBREVIVE
A UM POUSO FORCADO NESTA VELOCI
DADE"
6500  HTAB 5: VTAB 10: PRINT "V
ALORES RELATIVOS AO POUSO"
6510  PRINT : PRINT "VELOCIDADE
 DO AR "; INT (AS);" M/S": PRIN
T "DISTANCIA = "; INT ( SQR (PY
 * PY + PX * PX));" M": PRINT "
INCLINACAO FRONTAL = ";PT
6520  PRINT "INCLINACAO LATERAL
 = ";RL: PRINT "ROTACAO DO MOTO
R = "; INT (1000 * TC);" RPM"
6530  PRINT "DESVIO DO EIXO DA
PISTA = ";PX;" M": PRINT "DIREC
AO = ";AD;" GRAUS"
6540  PRINT : PRINT "QUER VOAR
NOVAMENTE (S/N)?"
6550  GET AS: IF AS = "S" THEN
 RUN
6560  IF AS = "N" THEN  HOME :
END
```

O laço principal do programa vai da linha 5500 à linha 5530. Inicialmente, ele chama a sub-rotina de controle do aparelho pelo teclado, que começa na linha 3000 e vai até a 3070. A linha 3000 verifica se alguma tecla foi apertada; as linhas 3010 e 3020 detectam as teclas Q e A, que diminuem e aceleram a rotação do motor; as linhas 3030 e 3040 descobrem as teclas P e L, que controlam a ascensão e a queda do aparelho; finalmente, as linhas 3050 e 3060 detectam as teclas Z e X, que viram o avião.

As linhas 5020 e 5070 permitem variar o nível de dificuldade da aterrissagem por intermédio da escolha da força e da inclinação do vento. A opção mais fácil é uma velocidade de 1 m/s numa direção que faz um ângulo de zero grau com o aparelho. A partir dos valores relacionados, a linha 5060 calcula o componente frontal da velocidade do vento, e a linha 5070, seu componente lateral. **WX** e **WY** são usados para alterar a posição do aparelho.

Se a linha 5510 perceber que o avião atingiu o solo, o programa prosseguirá na linha 6000. Ali serão verificados os valores das inclinações frontal e lateral, assim como os da velocidade do vento. Se todos eles estiverem dentro dos limites aceitáveis, a linha 6020 informará ao piloto que houve uma aterrissagem bem-sucedida.

Como no caso anterior, nem tudo estará perdido se aterrissarmos fora da pista. Caso a velocidade de pouso seja inferior a 40 m/s, a linha 6210 dirá que os danos foram pequenos. Se ela estiver entre 40 e 80 m/s, a linha 6220 dará conta de pequenos danos e alguns feridos. Se a velocidade for maior que 80 m/s, só nos restará procurar a caixa-preta entre os escombros: neste caso, o desastre terá sido total.

As linhas 6500 a 6530 mostram os valores de todas as variáveis no momento da aterrissagem. Estudando bem esses números, você aprenderá a pousar cada vez melhor.

As linhas 6540 a 6570, finalmente, nos dão a opção de jogar novamente.

Na última seção vimos que, se o jogo se estender por muito tempo, a tela gráfica pode ser corrompida por restos de cordões que se acumulam na memória. Se você tiver esse tipo de problema, pode corrigi-lo, modificando a linha 2000 para:

```
2000 X =  FRE (0): HCOLOR= 0
```

O comando **FRE(0)** limpa a área de cordões, evitando que ela invada a tela gráfica. No entanto, ele torna o programa mais lento.

A velocidade do programa é comprometida principalmente pela execução da sub-rotina que escreve na tela gráfica. Como os valores que devem ser constantemente atualizados ficam na porção inferior da tela, podemos acelerar consideravelmente o programa se liberarmos as quatro linhas inferiores para texto (veja o artigo *Os Comandos PEEK e POKE* na página 261) e usarmos **HTAB**, **VTAB** e **PRINT** para escrever ali esses valores.

Aqui estão as modificações para que o programa funcione no TK-2000.

```
10  FOR I = 0 TO 6
20  READ A: POKE 768 + I,A
30  NEXT
40  DATA  32,67,240,141,32,3,96
3000  POKE 800,0: CALL 768:K =
 PEEK (800)
```

As quatro primeiras linhas colocam uma pequena rotina em código de máquina na memória do TK-2000. Quando chamada pela linha 3000, essa rotina permite que se faça leitura do teclado, com auto-repetição.

A baixa velocidade de execução do programa se deve aqui às características específicas do TK-2000.

Antes de executar o programa, apague a linha 5505.

```
3000 IF PEEK(337)=255 THEN RETU
RN
3010 IF PEEK(341)=251 AND TC>.2
 THEN TC=TC-.2
3020 IF PEEK(344)=254 AND TC<8.
8 THEN TC=TC+.2
3030 IF PEEK(341)=247 THEN PT=P
T+1
3040 IF PEEK(342)=223 THEN PT=P
T-1
3050 IF PEEK(343)=223 AND RL>-3
0 THEN RL=RL-1
3060 IF PEEK(344)=247 AND RL<30
 THEN RL=RL+1
3070 RETURN
5020 CLS:INPUT"VELOCIDADE DO VE
NTO (1-50) M/S";X0
5025 IF X0>50 OR X0<1 THEN 5020
5030 INPUT"DIRECAO DO VENTO (0-
359) GRAUS";X1
5035 IF X1>359 OR X1<0 THEN 503
0
5040 X0=X0/3
5050 PRINT:PRINT"VELOCIDADE DO
VENTO= ";3*X0;"M/S":PRINT"DIREC
AO= ";X1;"GRAUS"
5060 WY=X0*COS(X1*C)
5070 WX=-X0*SIN(X1*C)
5500 GOSUB 3000:GOSUB 1000
5510 IF PZ<=0 THEN 6000
5520 GOSUB 2000
5530 GOTO 5500
6000 IF ABS(RL)>RT OR PT>TP OR
PT<0 OR VV>80 THEN 6100
6010 IF ABS(PX)>WR OR ABS(PY)>1
```

```
000 THEN 6200
6020 CLS:PRINT" PARABENS!    AT
ERRISSAGEM BEM   SUCEDIDA.":GOT
O 6500
6100 LINE(0,0)-(RND(128)+63,RND
(96)+48),PSET:LINE-(255,191),PS
ET:LINE(255,0)-(RND(128)+63,RND
(96)+48),PSET:LINE -(0,191),PSE
T
6110 FOR K=1 TO 2000:NEXT
6120 CLS:PRINT" LAMENTAMOS INFO
RMAR QUE O VOO   PJ26 SOFREU UM
 GRAVE ACIDENTE."
6140 PRINT" NAO HA SOBREVIVENTE
S":GOTO 6500
6200 CLS:PRINT" VOCE POUSOU FOR
A DA PISTA"
6210 IF VV<40 THEN PRINT" COMO
SUA VELOCIDADE ERA BAIXA,  QUAS
E NADA ACONTECEU":GOTO 6500
6220 IF VV<80 THEN PRINT" ENTRE
 MORTOS E FERIDOS,          SALV
ARAM-SE TODOS":GOTO 6500
6230 PRINT" NINGUEM SOBREVIVE A
 UM POUSO    NESTA VELOCIDADE!"
6500 PRINT:PRINT " VALORES RELA
TIVOS AO POUSO"
6510 PRINT:PRINT" VELOCIDADE DO
 AR =";INT(VV);"M/S":PRINT" DIS
TANCIA =";INT(SQR(PY*PY+PX*PX))
:PRINT" INCLINACAO FRONTAL =";P
T
6520 PRINT" INCLINACAO LATERAL
=";RL:PRINT" ROTACAO DO MOTOR =
";INT(10*TC)/10;" X 1000"
6530 PRINT" DESVIO DO EIXO DA P
ISTA =";INT(ABS(PX));"M":PRINT"
 DIRECAO =";AD;"GRAUS"
6540 PRINT:PRINT" QUER VOAR NOV
AMENTE (S/N) ?"
6550 A$=INKEY$:IF A$<>"S" AND A
$<>"N" THEN 6550
6560 IF A$="N" THEN CLS:END
6570 RUN
```

O laço principal vai de 5500 a 5530. Ele chama a sub-rotina de controle do aparelho pelo teclado, que vai de 3000 a 3070. A linha 3000 verifica se alguma tecla foi apertada; 3010 e 3020 detectam as teclas F e S, que diminuem e aceleram a rotação do motor; 3030 e 3040 descobrem as setas "para cima" e "para baixo", que controlam a ascensão e a queda do aparelho; 3050 e 3060 detectam as setas "direita" e "esquerda", que viram o avião.

As linhas 5020 e 5070 permitem variar o nível de dificuldade do pouso, escolhendo a força e a inclinação do vento. A melhor opção é uma velocidade de 1 m/s numa direção que faz um ângulo de zero grau com o aparelho. A linha 5060 calcula o componente frontal da velocidade do vento, e a 5070, seu componente lateral. **WX** e **WY** servem para alterar a posição do aparelho.

Se a linha 5510 perceber que o avião atingiu o solo, o programa prosseguirá na linha 6000. Ali serão verificados os valores das inclinações frontal e lateral, bem como da velocidade do vento. Se todos eles estiverem dentro dos limites aceitáveis, a linha 6020 informará ao piloto que houve uma aterrissagem bem-sucedida.

Uma aterrissagem fora da pista não implica, necessariamente, a destruição do aparelho. Assim, se a velocidade de pouso for inferior a 40 m/s, a linha 6210 dirá que os danos foram pequenos. Se ela estiver entre 40 e 80 m/s, a linha 6220 informará que alguma coisa foi destruída e que há alguns feridos. Se a velocidade for maior que 80 m/s, só nos restará procurar a caixa-preta entre os destroços do avião: o desastre terá sido total.

As linhas 6500 a 6530 mostram os valores das variáveis no momento do pouso. Estudando bem esses números, você aprenderá a pousar cada vez melhor.

Finalmente, as linhas 6550 a 6570 oferecem a opção de jogar de novo.

UM RELÓGIO NA TELA

Se você não tiver nada para fazer numa tarde de chuva, rompa o tédio executando a rotina apresentada neste artigo e veja surgir um relógio digital na tela do computador.

Usado para regular as atividades do micro, o relógio interno funciona a uma velocidade constante. Em alguns casos, seu conteúdo só pode ser lido e manipulado por instruções em código de máquina. Em outros, podemos cronometrar eventos através de instruções do BASIC, que variam segundo o modelo da máquina: **PAUSE** no Spectrum, **TIME** no MSX, **TIMER** no TRS-Color etc.

Tais instruções fazem com que o micro conte em unidades entre cem e cinqüenta avos de segundo, conforme a velocidade do relógio. Muitas operações usam o relógio de maneira similar: por exemplo, se o micro for programado para tocar música, temos que especificar a duração de cada nota.

CONTANDO O TEMPO

De fato, ainda que certas funções não tenham sua duração definida dessa maneira, o computador é um contador de tempo por excelência, executando os programas com o olho no relógio. Podemos, assim, fazer com que o computador conte o tempo para nós: basta escrever um programa em BASIC contendo um laço que imprima a hora, provoque uma pausa, imprima a nova hora, e assim sucessivamente. Se fizermos isso, logo veremos que a pausa deve ser um pouco menor que um segundo, pois o micro leva algum tempo para fazer as adições e impressões necessárias.

Um relógio desse tipo tem duas grandes desvantagens. A primeira é que ele só conta o tempo enquanto o computador estiver ligado. Mais grave do que esta, a segunda limitação consiste no fato de que, sempre que quisermos usar o micro para alguma outra coisa, teremos que desligar o relógio, pois não é possível executar dois programas BASIC ao mesmo tempo. O "desligamento" do relógio caracteriza, portanto, uma interrupção da UCP. Esse tipo de manipulação em geral exige um conhecimento maior da arquitetura do microcomputador, bem como o uso de instruções de acesso a recursos que só existem em linguagem de máquina.

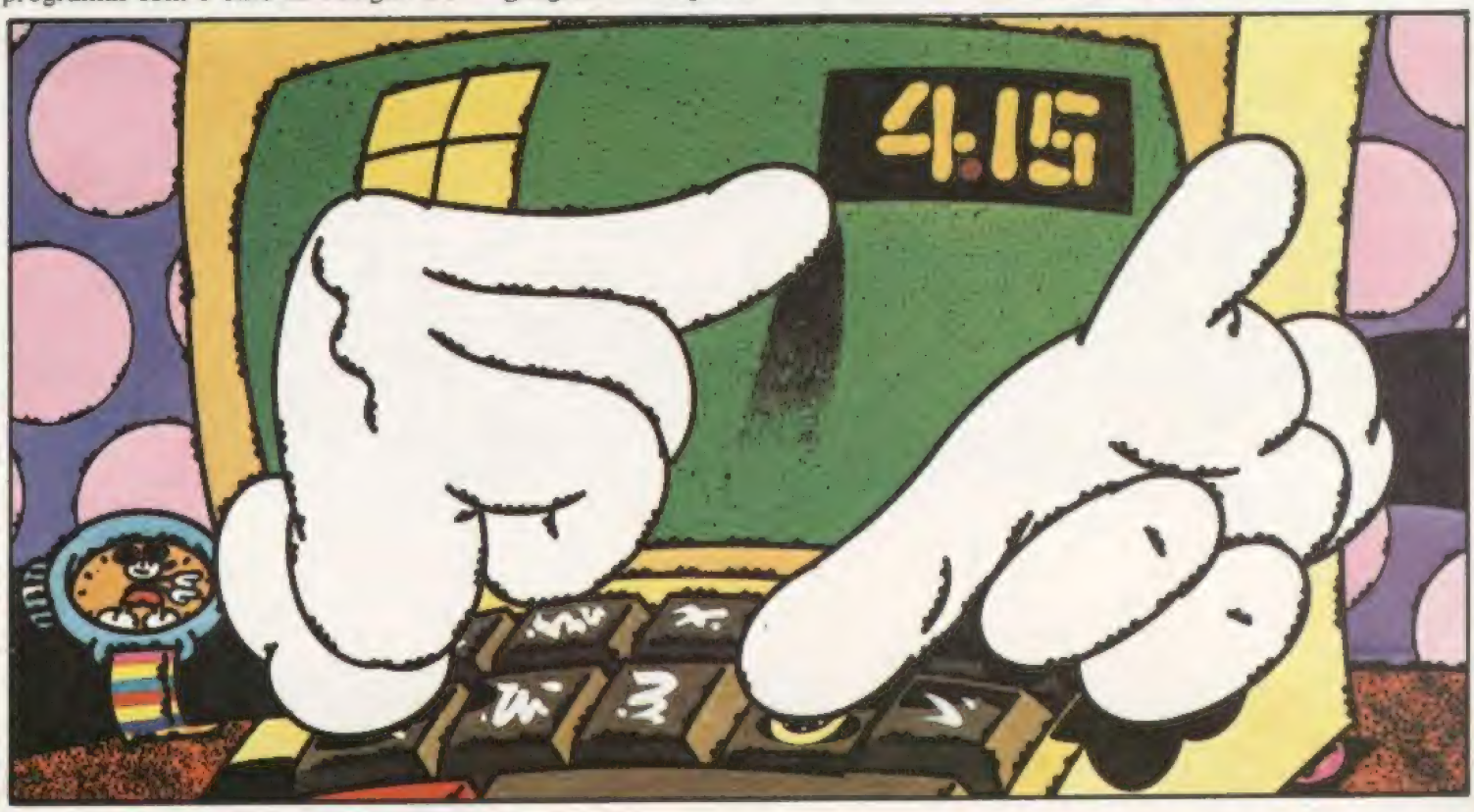
UM RELÓGIO INTERMITENTE

Para que um programa funcione simultaneamente com a execução de um BASIC, é preciso que ele contenha rotinas controladas por interrupção.

Os usuários do Spectrum já tiveram um exemplo desse tipo de rotina. De fato, o método utilizado é semelhante em todos os micros.

Quando está ligado, o micro é constantemente interrompido por uma fração de segundo, a intervalos regulares. Isso acontece mesmo que um programa em BASIC esteja sendo executado, pois o computador precisa "saber" se alguma tecla foi pressionada. Assim, o programa BASIC é interrompido enquanto o teclado é verificado, recomeçando novamente até a interrupção seguinte.

Durante a verificação do teclado, é possível introduzir uma rotina em código, de modo a executá-la nas "brechas" abertas pela interrupção do BASIC. O resultado é que os programas parecerão funcionar simultaneamente.



- O COMPUTADOR COMO RELÓGIO
- CONTANDO O TEMPO
- COMO FAZER INTERRUPÇÕES
- O MOSTRADOR DO RELÓGIO
- UM RELÓGIO INTERMITENTE
- UMA ROTINA EM CÓDIGO
- LIGUE O RELÓGIO
- ACERTE OS PONTEIROS
- **SAVE, LOAD, SOUND, PLAY**

Já que o processo de interrupção é controlado pelo relógio do micro, convém acertar o passo do relógio simplesmente contando o número de interrupções. A freqüência de interrupções varia de micro para micro, mas o princípio é o mesmo.

O programa que se segue simula um relógio no Spectrum e no TRS-Color, com horas, minutos e segundos contados a partir do momento em que começa a ser executado. Podemos acertá-lo de modo a fazê-lo funcionar, seja como relógio, seja como cronômetro. Esses valores são mostrados continuamente no canto superior direito da tela. O relógio permanecerá funcionando na tela, mesmo que outro programa esteja sendo executado ou digitado.

Como você não tardará a perceber, ele não prima muito pela precisão. Essa falha, porém, não se deve à rotina que acerta a hora, pois os atrasos provocados por ela são insignificantes (da ordem de milionésimos de segundo). Assim, o erro vem mais do relógio interno do que de qualquer outra fonte. Por outro lado, funções como **SAVE, LOAD, SOUND** e **PLAY** param o relógio enquanto estiverem sendo executadas. A hora é exibida, num mostrador digital, no canto superior direito do vídeo, apagando qualquer coisa impressa ali anteriormente. Se isso causar problema, reorganize sua tela a fim de evitar a primeira linha. Por exemplo, você poderia colocar a hora digital no canto superior (esquerdo ou direito), desde que o modelo do seu micro o permitisse. Neste caso, o programa em linguagem de máquina teria que ser alterado.

S

O programa a seguir funciona tanto na versão de 16K, como na de 48K do Spectrum. Antes de começar a rodá-lo, tome alguns cuidados especiais, pois cartuchos e expansões conectados ao micro podem dificultar sua execução. É o caso, por exemplo, de interfaces para impressora, joysticks, etc.

```
10 CLEAR 32319: LET total=0
20 FOR n=32320 TO 32554: READ
a: POKE n,a: LET total=total+
a: NEXT n
30 IF total<>24216 THEN
PRINT "Erro nos dados": STOP
40 RAND USR 32320
50 DATA 33,0,0,34,120,92,34,
121,92,62,40,237,71,237,94,
201,0,64,0,0
60 DATA 62,62,237,71,237,86,
201,0,229,213,197,245,58,91,
126,60,50,91,126,254
70 DATA 50,32,50,175,50,91,
126,58,120,92,60,50,120,92,
254,60,32,35,175,50
80 DATA 120,92,58,121,92,60,
50,121,92,254,60,32,20,175,50
,121,92,58,122,92
90 DATA 60,50,122,92,254,13,
32,5,62,1,50,122,92,58,122,92
,38,0,111,17
100 DATA 23,64,205,234,126,58,
121,92,38,0,111,17,26,64,205,
234,126,58,120,92
110 DATA 38,0,111,17,29,64,205
,234,126,17,208,61,33,29,64,
205,34,127,17,208
120 DATA 61,33,26,64,205,34,
127,62,120,33,24,88,119,17,25,
88,1,7,0,237
130 DATA 176,205,191,2,241,193
,209,225,251,201,237,83,80,126
,1,246,255,205,251,126
140 DATA 1,255,255,205,251,126
,201,175,9,60,56,252,237,66,61
,198,48,229,205,21
150 DATA 127,33,80,126,52,42,
80,126,205,34,127,225,201,237,
75,54,92,38,0,111
160 DATA 41,41,41,9,235,201,6,
8,26,119,36,19,16,250,201
```

Essa rotina é formada por números distribuídos em linhas **DATA**; eles são colocados na memória pelo BASIC, usando **POKE**. Tal quantidade de números propicia erros de digitação; assim, a linha 20 verifica a soma dos números: se o resultado não corresponder ao esperado, a linha 30 interromperá a execução do programa com uma mensagem de erro, para que você verifique as linhas **DATA**. Esse método é chamado de "checagem por soma" e é recomendável quando há muitos números.

A linha 10 protege o topo da memória para colocar ali a rotina em código, que é executada pela linha 40 . O relógio começa com 00:00:00, mas podemos acertá-lo com:

```
POKE 23672, (segundos)
POKE 23673, (minutos)
POKE 23674, (horas)
```

Os números nos comandos **POKE** devem estar no intervalo permitido — de 0 a 60 para minutos e segundos, e de 1 a 12 para horas. Se quisermos "zerar" o relógio, é mais fácil usar:

```
RAND USR 32320
```

que executa a rotina novamente. Você vai precisar desse comando, se o programa em BASIC for apagado por **NEW**.

T

Essa rotina não funciona com drives de disquete ligados ao micro.

```
10 CLEAR 200,32599
20 FOR J=32600 TO 32679
30 READ N
40 POKE J,N
50 NEXT
100 DATA 204,0,0,253,127,252,25
3,127,254,48,140,4,191,1,13,57
110 DATA 206,127,164,142,128,0,
166,130,76,161,192,38,9,111,132
120 DATA 140,127,252,38,242,134
,1,167,132,206,4,32,142,127,255
130 DATA 79,230,130,192,10,45,3
,76,32,249,195,47,58,237,195,17
140 DATA 131,4,25,47,6,134,58,1
67,194,32,229,126,137,76,50,60,
60,13
```

A rotina em código está nas linhas **DATA**, sendo colocada na memória pelo **POKE** da linha 40. Qualquer erro nas linhas **DATA** será fatal; portanto, grave o programa antes de executá-lo. Verifique os números cuidadosamente. Um erro muito comum é esquecer alguma vírgula, substituir uma vírgula por um ponto ou vice-versa. Tudo isto prejudica a seqüência de leitura do comando **READ**. Execute o programa para carregar a rotina em código. Para ligar o relógio, use:

```
EXEC 32600
```

Isso inicia a atividade do relógio com 00:00:00. Para acertá-lo, digite em modo direto, pelo teclado:

```
POKE 32764, (horas)
POKE 32765, (minutos)
POKE 32766, (segundos)
```

Não empregue valores impróprios para horas, minutos e segundos: o computador não tem meios de checá-los.

SPRITES PARA O TRS-80 (2)

- CORRESPONDÊNCIA ENTRE CÓDIGO E CARACTERE GRÁFICO
- COMO CALCULAR O CÓDIGO
- BITS QUE SE ACENDEM
- APLICAÇÕES

Aprenda a empregar os caracteres gráficos pré-programados do TRS-80 em desenhos mais complexos e capacite-se a trabalhar com os úteis "sprites programáveis".

Se examinarmos com cuidado os caracteres gráficos do TRS-80, que ocupam a faixa de códigos que vai de 129 a 193 (veja tabela no capítulo anterior desta lição), podemos notar facilmente que eles são formados por todas as combinações possíveis de seis *pixels* retangulares, acesos ou apagados. Portanto, a matriz de um caractere tem esta forma:

Você verá agora que não é necessário memorizar a tabela de caracteres gráficos do TRS-80, nem consultá-la a todo momento ao se projetar um desenho mais complexo, formado por vários desses caracteres. A razão é que existe uma relação matemática bem definida entre a forma do gráfico e o seu código numérico. Ou seja, os códigos não foram atribuídos ao acaso, como para computadores de outras linhas.

BITS QUE SE ACENDEM

Todo caractere gráfico é formado pela combinação de seis pixels — dois na horizontal e três na vertical —, que podem estar individualmente acesos ou apagados. Cada um deles corresponde exatamente ao pontinho de luz que pode ser ligado ou desligado com os comandos gráficos **SET** e **RESET**.

Por quê? Sabemos que a tela de baixa resolução do TRS-80 tem 128 pontos na horizontal (ou seja, 64 caracteres vezes 2 pixels), e 48 na vertical (isto é, 16 linhas vezes 3 pixels).

Cada pixel corresponde a um bit na memória de vídeo do micro. Como cada byte da memória tem oito bits, e cada caractere seis bits, sobram dois bits não usados para representar pixels. Um byte da memória de vídeo no TRS-80 poderia ser representado assim:

0	1
2	3
4	5

7	6	5	4	3	2	1	0

Os bits numerados de 0 a 5 correspondem aos seis pixels habituais do caractere gráfico.

O bit 6 não é usado para representação de caracteres gráficos, e o bit 7 é sempre igualado a 1, para todos os caracteres gráficos.

Por exemplo, se quisermos obter o código gráfico 134, precisaremos acender sucessivamente 1 e 2:

7	6	5	4	3	2	1	0
1		0	0	0	1	1	0

O código 134 é obtido fazendo-se a conversão de binário para decimal:

bit		valor	multiplicador		resultado
0	0	x	1	=	0
1	1	x	2	=	2
2	1	x	4	=	4
3	0	x	8	=	0
4	0	x	16	=	0
5	0	x	32	=	0
6	0	x	64	=	0
7	1	x	128	=	128
Soma					134

Dessa maneira, é relativamente fácil calcular o valor do código de qualquer caractere. Para isso, basta verificar as posições ocupadas pelos diversos pixels "acesos" na matriz do caractere, adicionar em seguida os valores inscritos na posição, e somar 128 ao resultado (daí a razão de o número 128 ser considerado também como um código gráfico):

1	2
4	8
16	32

APLICAÇÕES

Este fato tem diversas implicações: é muito fácil, por exemplo, inverter-se os gráficos escritos na tela (para isso, basta transformar os bits da memória gráfica, de modo que os que são 1 passem para 0, e vice-versa). É possível obter outras transformações matemáticas com os códigos gráficos, com resultados curiosos.

Mas a aplicação que mais nos interessa no momento relaciona-se com a definição de figuras complexas. Neste caso, o primeiro passo consiste em armazenar os códigos numéricos dos caracteres gráficos que compõem tais figuras, em uma linha **DATA**, que é lida e armazenada (concatenada) em uma variável alfanumérica. Para economizar espaço, somente o valor diminuído de 128 é armazenado.

Tomemos como exemplo o programa a seguir; ele define uma variável **A$** que contém a imagem de um aeroplano simples, formado pelos caracteres 141, 170, 174, 170 e 140.

```
10 CLS:A$=""
20 FOR I=1 TO 5:READ N
30 A$=A$+CHR$(128+N)
40 NEXT I
50 DATA 13,42,46,42,12
```

O aviãozinho "voa" do seguinte modo: impresso inicialmente em uma posição da tela (com **PRINT @**), ele é sucessivamente apagado com cinco caracteres em branco e exibido na posição seguinte, e assim por diante. Acrescente estas linhas e rode o programa:

```
60 FOR I=256 TO 313
70 PRINT @ I,A$;
80 FOR J=1 TO 40:NEXT
90 PRINT @ I,"     ";
100 NEXT I
```

Portanto, **A$** é, de certa forma, uma espécie de sprite.

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR LINHA

LINHA	FABRICANTE	MODELO
Apple II +	Appletronica	Thor 2010
Apple II +	CCE	MC-4000 Exato
Apple II +	CPA	Absolutus
Apple II +	CPA	Polaris
Apple II +	Digitus	DGT-AP
Apple II +	Dismac	D-8100
Apple II +	ENIAC	ENIAC II
Apple II +	Franklin	Franklin
Apple II +	Houston	Houston AP
Apple II +	Magnex	DM II
Apple II +	Maxitronica	MX-2001
Apple II +	Maxitronica	MX-48
Apple II +	Maxitronica	MX-64
Apple II +	Maxitronica	Maxitronic I
Apple II +	Microcraft	Craf II Plus
Apple II +	Milmar	Apple II Plus
Apple II +	Milmar	Apple Master
Apple II +	Milmar	Apple Senior
Apple II +	Omega	MC-400
Apple II +	Polymax	Maxxi
Apple II +	Polymax	Poly Plus
Apple II +	Spectrum	Microengenho I
Apple II +	Spectrum	Spectrum ed
Apple II +	Suporte	Venus II
Apple II +	Sycomig	SIC I
Apple II +	Unitron	AP II
Apple II +	Victor do Brasil	Elppa II Plus
Apple II +	Victor do Brasil	Elppa Jr.
Apple IIe	Microcraft	Craft IIe
Apple IIe	Microdigital	TK-3000 IIe
Apple IIe	Spectrum	Microengenho II
MSX	Gradiente	Expert GPC-1
MSX	Sharp	Hotbit HB-8000
Sinclair Spectrum	Microdigital	TK-90X
Sinclair Spectrum	Timex	Timex 2000
Sinclair ZX-81	Apply	Apply 300
Sinclair ZX-81	Engebras	AS-1000
Sinclair ZX-81	Filcres	NEZ-8000
Sinclair ZX-81	Microdigital	TK-82C
Sinclair ZX-81	Microdigital	TK-83
Sinclair ZX-81	Microdigital	TK-85
Sinclair ZX-81	Prologica	CP-200
Sinclair ZX-81	Ritas	Ringo R-470
Sinclair ZX-81	Timex	Timex 1000
Sinclair ZX-81	Timex	Timex 1500
TRS-80 Mod. I	Dismac	D-8000
TRS-80 Mod. I	Dismac	D-8001/2
TRS-80 Mod. I	LNW	LNW-80
TRS-80 Mod. I	Video Genie	Video Genie I
TRS-80 Mod.III	Digitus	DGT-100
TRS-80 Mod.III	Digitus	DGT-1000
TRS-80 Mod.III	Kemitron	Naja 800
TRS-80 Mod.III	Prologica	CP-300
TRS-80 Mod.III	Prologica	CP-500
TRS-80 Mod.III	Sysdata	Sysdata III
TRS-80 Mod.III	Sysdata	Sysdata Jr.
TRS-80 Mod.IV	Multix	MX-Compacto
TRS-80 Mod.IV	Sysdata	Sysdata IV
TRS-Color	Codimex	CS-6508
TRS-Color	Dynacom	MX-1600
TRS-Color	LZ	Color 64
TRS-Color	Microdigital	TKS-800
TRS-Color	Prologica	CP-400

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR NOME

FABRICANTE	MODELO	PAÍS	LINHA
Appletronica	Thor 2010	Brasil	Apple II +
Apply	Apply 300	Brasil	Sinclair ZX-81
CCE	MC-4000 Exato	Brasil	Apple II +
CPA	Absolutus	Brasil	Apple II +
CPA	Polaris	Brasil	Apple II +
Codimex	CS-6508	Brasil	TRS-Color
Digitus	DGT-100	Brasil	TRS-80 Mod.III
Digitus	DGT-1000	Brasil	TRS-80 Mod.III
Digitus	DGT-AP	Brasil	Apple II +
Dismac	D-8000	Brasil	TRS-80 Mod. I
Dismac	D-8001/2	Brasil	TRS-80 Mod. I
Dismac	D-8100	Brasil	Apple II +
Dynacom	MX-1600	Brasil	TRS-Color
ENIAC	ENIAC II	Brasil	Apple II +
Engebras	AS-1000	Brasil	Sinclair ZX-81
Filcres	NEZ-8000	Brasil	Sinclair ZX-81
Franklin	Franklin	USA	Apple II +
Gradiente	Expert GPC1	Brasil	MSX
Houston	Houston AP	Brasil	Apple II +
Kemitron	Naja 800	Brasil	TRS-80 Mod.III
LNW	LNW-80	USA	TRS-80 Mod. I
LZ	Color 64	Brasil	TRS-Color
Magnex	DM II	Brasil	Apple II +
Maxitronica	MX-2001	Brasil	Apple II +
Maxitronica	MX-48	Brasil	Apple II +
Maxitronica	MX-64	Brasil	Apple II +
Maxitronica	Maxitronic I	Brasil	Apple II +
Microcraft	Craft II Plus	Brasil	Apple II +
Microcraft	Caft IIe	Brasil	Apple IIe
Microdigital	TK-3000 IIe	Brasil	Apple IIe
Microdigital	TK-82C	Brasil	Sinclair ZX-81
Microdigital	TK-83	Brasil	Sinclair ZX-81
Microdigital	TK-85	Brasil	Sinclair ZX-81
Microdigital	TK-90X	Brasil	Sinclair Spectrum
Microdigital	TKS-800	Brasil	TRS-Color
Milmar	Apple II Plus	Brasil	Apple II +
Milmar	Apple Master	Brasil	Apple II +
Milmar	Apple Senior	Brasil	Apple II +
Multix	MX-Compacto	Brasil	TRS-80 Mod.IV
Omega	MC-400	Brasil	Apple II +
Polymax	Maxxi	Brasil	Apple II +
Polymax	Poly Plus	Brasil	Apple II +
Prologica	CP-200	Brasil	Sinclair ZX-81
Prologica	CP-300	Brasil	TRS-80 Mod.III
Prologica	CP-400	Brasil	TRS-Color
Prologica	CP-500	Brasil	TRS-80 Mod.III
Ritas	Ringo R-470	Brasil	Sinclair ZX-81
Sharp	Hotbit HB-8000	Brasil	MSX
Spectrum	Microengenho I	Brasil	Apple II +
Spectrum	Microengenho II	Brasil	Apple IIe
Spectrum	Spectrum ed	Brasil	Apple II +
Suporte	Venus II	Brasil	Apple II +
Sycomig	SIC I	Brasil	Apple II +
Sysdata	Sysdata III	Brasil	TRS-80 Mod.III
Sysdata	Sysdata IV	Brasil	TRS-80 Mod.IV
Sysdata	Sysdata Jr.	Brasil	TRS-80 Mod.III
Timex	Timex 1000	USA	Sinclair ZX-81
Timex	Timex 1500	USA	Sinclair ZX-81
Timex	Timex 2000	USA	Sinclair Spectrum
Unitron	AP II	Brasil	Apple II +
Victor do Brasil	Elppa II Plus	Brasil	Apple II +
Victor do Brasil	Elppa Jr.	Brasil	Apple II +
Video Genie	Video Genie I	USA	TRS-80 Mod. I

UM LOGOTIPO PARA CADA MODELO DE COMPUTADOR

INPUT foi especialmente projetado para microcomputadores compatíveis com as sete principais linhas existentes no mercado.
Os blocos de textos e listagens de programas aplicados apenas a determinadas linhas de micros podem ser identificados por meio dos seguintes símbolos:

Sinclair ZX-81

TRS-80

TK-2000

MSX

Spectrum

TRS-Color

Apple II

Quando o emblema for seguido de uma faixa, então tanto o texto como os programas que se seguem passam a ser específicos para a linha indicada.

NO PRÓXIMO NÚMERO

PROGRAMAÇÃO BASIC
Simule a trajetória de uma bola no computador, utilizando modelos matemáticos em seus programas.

PROGRAMAÇÃO DE JOGOS
Prepare-se para faiscar ouro no duro chão de Serra Pelada. E fique rico do dia para a noite.

PROGRAMAÇÃO BASIC
Como utilizar caracteres gráficos dentro de um programa. Uma tabela de referência.

CÓDIGO DE MÁQUINA
Faça algumas alterações no programa Assembler para o MSX e adapte-o para os micros da linha TRS-80.

PROGRAMAÇÃO BASIC
Como armazenar figuras mais complexas em uma única variável.